Logistics Management Institute

Better Development Cost Estimating Through Modeling and Simulation

PA903T1

May 2000

Gerald J. Belcher
John Dukovich
Dr. David Lee
Bobby Jackson

LMI®

Better Development Cost Estimating Through Modeling and Simulation

PA903T1/MAY 2000

Executive Summary

More than half of all Major Defense Acquisition Programs historically experience cost growth in excess of 20 percent, and a large portion of those realize more than 50 percent growth. Cost growth gives rise to numerous problems. In today's fiscal environment, when the government pays much more than forecasted for the development of a critical product, some other fiscal priority must go unfulfilled. Or, if the cost growth is viewed as extreme, even an important program may be cancelled. Clearly, it is very important to estimate program costs accurately.

Our previous research established that models and methods the Department of Defense (DoD) acquisition and cost communities use contribute substantially to the cost growth problem. They do not capture some important aspects of contemporary development programs. Specifically, they routinely underestimate the complexity of the development task while not addressing activities such as reworking and retesting after failed evaluations.

During this research program, we explored ways of understanding the true causes of development costs in an effort to represent them better in our cost models. We also assessed some alternative analysis and estimation techniques aimed specifically at the shortcomings of extant methods. While we present no specific models or tools here, we do evaluate the usefulness of several estimating approaches.

In Chapter 2 of this report, we explore the analytical appropriateness of using activity networks, or cost-time diagrams, to analyze cost and schedule for large-scale development programs. We highlight the complexities involved in accurately representing a major program or project. We conclude that using a class of activity networks, most notably in a simulation like the Graphical Evaluation and Review Technique Simulation, holds significant promise as a predictor of a program's cost and schedule while capturing the interrelationships inherent in development programs. We conclude also that modeling complex activity networks would require powerful, and likely costly, programming applications.

Also, we look at the use of simulation to model program execution and explicitly capture the complicated schedule, constraints, and factor interdependence that are so problematic for other analytical methods. We highlight the advantages and dis-

advantages of using a simulation for this purpose. We conclude that discrete-event simulation has the attributes required to effectively model the activities and effort in a typical development program, and to render improved cost estimates.

Finally, in Chapter 3, we suggest a way to proceed that should result in the development of an enhanced cost prediction tool. We describe a research and development strategy that designs, develops, tests, improves, and then fields a program execution simulation that should enhance program management and provide government acquisition organizations with robust cost and schedule forecasts for major development programs.

The implementation of these advanced modeling methods will require significant resources. We recommend that the development program be approached in at least three phases extending over multiple years. The first phase would focus on research, modeling, data collection, and prototyping. Subsequent phases would demonstrate the concept using an actual contemporary development program and then would implement the proven system DoD-wide.

The development program also would require a significant labor effort. We estimate that between 4.5 and 6 man-years of effort would be needed. There are also a number of other important considerations:

- Verification, validation, and accreditation of the system across diverse acquisition programs will be a significant issue and should be addressed at the outset of the program.
- The concept will require much more detailed contractor cost data than is currently collected. We recommend actions to obtain and use that data.
- Constant and widespread tailoring of the product will make configuration management critical to the system's success.
- Proponents of the system must develop incentives for contractors and project offices to encourage them to be open and to share the required information.

Contents

FIGURES

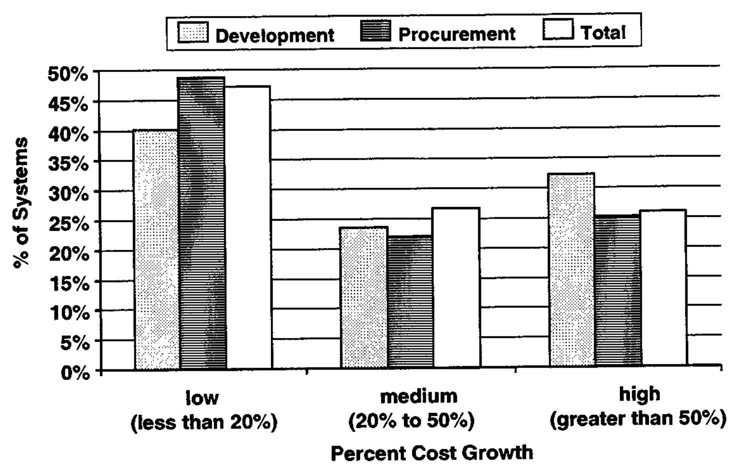
TABLES

Chapter 1
Introduction

BACKGROUND

The Department of Defense (DoD) does not do a good job of estimating development costs for its major defense acquisition programs. That bold assertion is supported by evidence that shows a majority of Major Defense Acquisition Programs (MDAP) experience significant development cost growth above their Milestone II cost estimates (see Figure 1-1).

Figure 1-1. Programs Experiencing Cost Growth



Source: Office of the Secretary of Defense, Program Analysis and Evaluation, Cost Analysis Improvement Group, October 1998.

Figure 1-1 shows that, historically, more than half of such programs experienced what we term medium or high cost growth—greater than 20 percent growth. Fully one-third of all the programs have seen more than 50 percent growth in development cost. Of course, cost growth gives rise to a number of problems. Clearly, in today's fiscal environment, when the government pays much more than forecast for a development, some other fiscal priority goes unfulfilled. Or, if the growth is viewed as extreme, an important program may be cancelled altogether.

The Cost Analysis Improvement Group (CAIG) in the Office of the Secretary of Defense (OSD) believes that the methods the DoD acquisition and cost analysis communities use to estimate development costs are, at least partially, responsible for this phenomenon because they fail to capture pertinent aspects of program execution that lead to significant program cost.

In 1998, to address this growing problem, the CAIG tasked the Logistics Management Institute (LMI) to determine how contemporary large-scale development programs incur cost and to assess the usefulness of existing DoD tools for estimating those costs. The CAIG hypothesized that recent technological and economic changes are so profound that they call into question the validity of the methods the cost analysis community uses to estimate the cost of new product developments for DoD. Specifically, dramatic consolidation of defense prime contractors, reformed acquisition practices in DoD, new high-technology products and production, and contractors' lowered expectations for large production runs of defense systems have rendered many of the current estimation tools all but obsolete. The CAIG contends that we must find better ways of estimating development costs. Whatever methods and models DoD uses to characterize product development should be evaluated against the backdrop of these new realities.

The purpose of this research program is to

- understand current product development processes,
- identify appropriate ways to estimate the cost of product development, and
- provide useful guidance that will help DoD cost analysts properly forecast the costs of development programs.

WHAT WE HAVE DONE TO DATE

During the first phase of this research program, we surveyed the most widely used methods of estimating development costs and found each of them lacking.[1] They all appear to be based on a few fundamental ideas. For example, in many product areas, cost analysts have long estimated the cost of developing an item as a multiple of the theoretical first unit cost of manufacturing the item. This method clearly is not sound for systems that require the wholesale integration of commercial-off-the-shelf (COTS) hardware into complex systems. Neither is it an appropriate technique when the development effort focuses primarily on creation of new intellectual content rather than on development of new hardware systems or components.

We also investigated other popular estimation methods. These include parametric estimation based on system and performance specifications or on performance parameter trends, should-cost methods, and decomposition and estimation by analogy. These techniques depend on how similar the developmental system is to some calibrating system. We found that estimates made using these methods were inaccurate in a majority of documented cases.

[1] Belcher, G. and D. Lee, *Estimating Development Costs in the Defense Electronics Industry,* LMI Report PA805T2, January 1999.

This report summarizes our research and analysis assessing the potential of some concepts little used in the field of cost estimating—Cause-and-Effect (C-E) analysis and project execution simulation. This research is aimed at determining what actually "causes" development cost and to finding improved estimating tools that address the concerns described above.

WHAT WE HAVE LEARNED SO FAR

During our earlier research, we examined the development of electronics products, specifically Global Positioning System (GPS) user equipment, with the goal of understanding the development process and development costs in a sector with significant civil influence. We also sought to capture and capitalize on commercial best practices for estimating those costs.

Our analysis of defense electronics programs revealed the following:

- Past estimates of development costs (for defense electronics) often were inaccurate because they
  - were based on overly optimistic, success-oriented schedules;
  - were based, inappropriately, on perfect matches of people with work;
  - did not include ways to adjust with technology trends; and
  - did not include ways to adjust to change.
- The information gained about development of GPS user equipment supports the basic assumption motivating this research—that advances in products and development processes and changes in the dynamics of the marketplace essentially have invalidated the current set of models for estimating development cost. The GPS information also leads us to think it likely that this is the case for the electronics industry as a whole, and may well affect other product sectors.
- A clearer analytic appreciation of the *factors* that drive modern product development is needed to understand how to address model shortcomings.

We found that one prevalent reason analogy-based methods fall short as predictors of cost is the aptness (or inaptness) of the explanatory variables they use. We asked ourselves, "Do these variables really characterize the actual cost drivers?" That question fueled our current research.

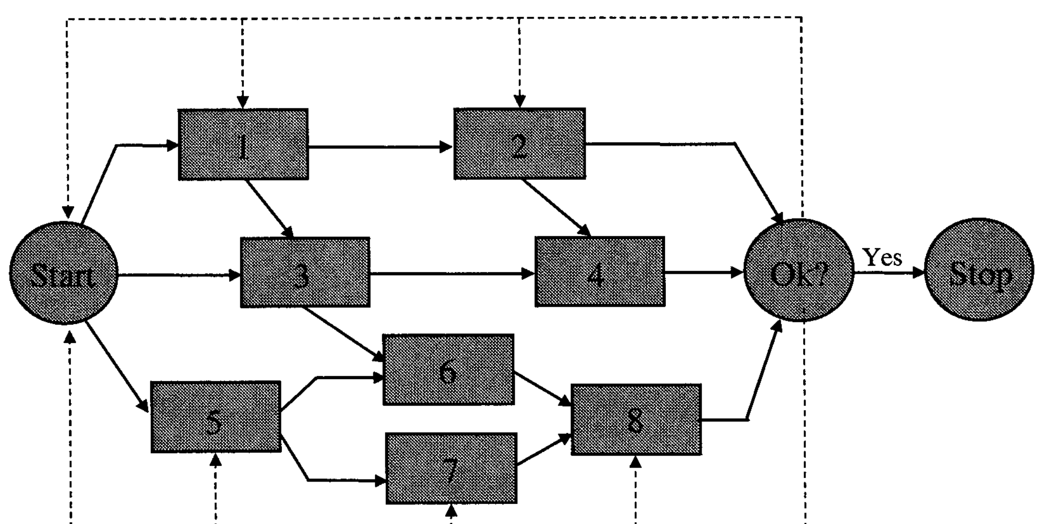
WHAT IS IN THIS REPORT

During the current phase of our research we devote significant effort to determine what drives development cost and what characterizes those drivers. That knowledge forms the foundation for our subsequent analyses. We used an approach that combines C-E analysis and Pareto analysis to assist in identifying the factors that contribute most to development cost in some large acquisition programs. We captured the methodology and its application to major missile defense programs in a letter report.[2] (The report appears in Appendix A.)

We found that among the factors poorly represented in current cost estimation models are time (schedule), program constraints, and the interdependencies of program events. Further, we concluded that cost models must define in detail the development activities involved in achieving a stressing technical requirement.

We wanted to use these findings and capitalize on the strengths of modeling program execution to address the shortcomings we have seen in current cost estimating techniques. Our CAIG sponsors suggested an approach to estimate the schedule and cost of a development program as outputs of a discrete event simulation. The simulation could be used to model the execution of each specific development program. The new approach should provide a means to evaluate schedule and cost risk, be flexible enough to be modified as changes occur, and its results should be intuitive. The model would account for funding constraints. Figure 1-2 shows a stylized representation of the model.

Figure 1-2. Generalized Program Execution Model



[2] Belcher, G. and J. Dukovich, *Improved Methods for Estimating Development Cost: Cause-and-Effect Analysis,* LMI Report PA903L1, January 2000.

The simulation would work through the program's various work packages in order of precedence. Completion of each work package would depend on a bivariate distribution of effort and time. The output of this stochastic model includes estimations of total program schedule and total program effort (which can be mathematically converted to total cost) and the estimated errors in schedule and cost.

The remainder of this report summarizes our research and analysis assessing this concept. Specifically, we take a close look at using activity networks and discrete-event simulation to further our goal of achieving improved development cost estimates.

Chapter 2

Program Execution Modeling

Overview

We began this phase of our research by investigating the efficacy of using C-E and influence diagramming to determine what actually *causes* development cost and how one might characterize the effects of these causes (see Appendix A). While we concluded that the techniques can provide some valuable insights into the development process and will aid in getting to the major roots of development cost, we recognize that there are still many very complicated contributors to cost that act on these drivers. These contributors might be quite different for each program. This uniqueness drives us to look at the alternative of modeling individual programs.

The DoD's management of development programs might be significantly improved by an effective, widely accepted method of modeling the programs' costs and completion times. Such a method should treat cost and completion time as dependent random variables, capturing the dispersion observed in these features. It should account for management's choices in assigning resources to the subprojects that make up a complete project.

The modeling method should reflect several constraints: precedence among subprojects, as well as overall and intermediate time and budget constraints. It should be sufficiently flexible to reflect changes in the project, and it should show how cost-time distributions become less dispersive as the project moves toward completion.

The method should represent all important features of a large, complex weapon system development program, but be simple enough that key queries, such as the overall variation of total cost and completion time, can be addressed with reasonable computational effort, involving analysis, simulation, or both.

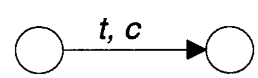
The class of models used should be so clearly useful to the entire DoD development community—program offices, the military departments' organizations for managing developments, OSD's oversight operations, and, particularly, the OSD CAIG—that nearly all program offices would develop, maintain, use, and share such a model with the community.

The OSD CAIG recognizes the benefits that the DoD acquisition community would enjoy if such a class of modeling methods were available. They note in particular how far such models would go in improving the generally unsatisfactory set of methods available for estimating the development costs of major defense acquisition programs. The CAIG asked LMI to assess the state of knowledge of program modeling and analysis. That assessment also includes an evaluation of the potential for the concept to provide better development cost estimates and provides a measure of its practicality for routine use.

This chapter contains our response. In it, we explore the possibility of estimating costs and completion times for development programs as dependent random variables, by decomposing a program into a set of subprograms for which analysts can usefully estimate individual cost-time distributions from data. We can use several representations or modeling methods for this. To avoid suggesting a specific one, we will refer to any decomposition representation for a given project as a cost-time diagram (CTD).

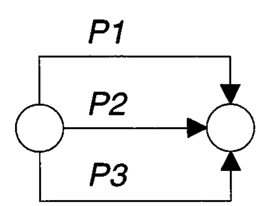
We note that some present methods of estimating development cost attempt to make estimates based on the simplest possible CTD. It has just one subprogram, that is, in fact, the entire project (Figure 2-1) with its associated cost *c* and completion time *t*.

Figure 2-1. Simplest Cost-Time Diagram



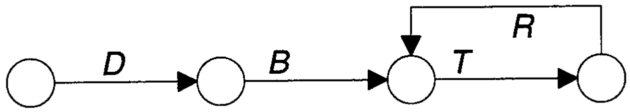
Even a slightly more detailed breakdown may give helpful insights. Some estimating methods decompose a project into a few major components (Figure 2-2). Decomposing an airplane development into separate estimates for airframe, avionics, and propulsion is an example of this.

Figure 2-2. Major-Component Decomposition



Further decomposition will, we believe, give substantial benefits. For example, separating a project or subproject into design, build, and test, with a feedback loop representing action after an unsuccessful test, shows how substantially quality control may affect a component's development time and cost (see Figure 2-3).

Figure 2-3. Project with Rework Feedback Loop



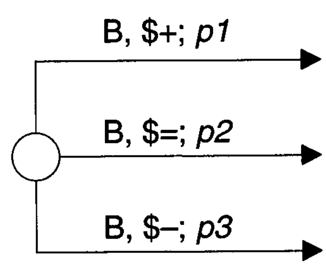
In Figure 2-3, let *D, B,* and *T* denote respectively the cost to design, build, and test a component, and let *R* represent the cost of rework after an unsuccessful test.[1] Let *p* be the probability of success on any test. Then the component's total cost *C* is a discrete random variable taking values C_n, $n = 0, 1, 2,...,$ where

$$C_n = D + B + (n+1)T + nR; \quad P(C_n) = p(1-p)^n. \qquad \text{[Eq. 2-1]}$$

It follows that, if testing costs 20 percent of the cost of building the developmental items, and rework after an unsuccessful test costs 10 percent of the building cost, and if the probability of a successful test is 0.5 on each try (an assumption that may be optimistic), there is a 14 percent chance that test and rework will increase the cost of building a successful item by 50 percent over the cost of building and testing one time. (One could argue that this model likely represents the experiences of a program like the Theater High Altitude Area Defense (THAAD) system. See Appendix A.)

CTDs also offer opportunities to treat some of the causes of development cost, identified in Appendix A, that are not commonly considered. For example, one can determine the impacts of political forces and military needs represented by the *urgency* and *funding constraints* entries of Figure A-4 with a CTD element of the kind shown in Figure 2-4.

Figure 2-4. CTD with Optional Arcs for Different Funding



In Figure 2-4, a "Build" phase will be executed with increased funding ($+), the planned funding ($=), or reduced funding ($–) with respective probabilities *p1, p2,* and *p3*.

[1] While the CTDs of Figure 2-1 and Figure 2-2 may be analyzed by classical CPM and PERT methods, the one in Figure 2-3 cannot. We will discuss more capable methods later.

Estimates based on CTDs such as those of Figure 2-3 and Figure 2-4 will generate reduced dispersion when made at later stages of a program's execution. In Figure 2-3, after the component passes the indicated test, the number of reworks is known, and the cost of building and testing this component is a known constant rather than a random variable. Similarly, in Figure 2-4, the branch taken eventually will be known, and subsequent estimates will take that fact into account.

A CTD supports several useful analyses. An important general analysis is, of course, to estimate the joint distribution of completion time and total cost. Specifying completion times for each subprogram leads to a distribution of total cost for fixed completion time. Versions of the "budget problem" (for given confidence that total cost will not exceed $B > 0$, find the distribution of completion time) and the "deadline problem" (for given confidence that completion time will not exceed $T > 0$, find the distribution of total cost) may be treated. For all but the simplest projects, analytic results may well be computationally infeasible. Approximations or simulations may help in complex cases.

This chapter provides a brief overview of methods for modeling development programs to analyze completion times and costs. Then, for the sake of completeness, and to introduce basic ideas in an uncluttered setting that many readers will find familiar, we briefly review deterministic time and cost-time representations. This part of the chapter relates the present work to the classical Critical Path Method (CPM), and it is a natural place to introduce the important idea of computational complexity. This term addresses a model's practicality. This section ends with descriptions of early deterministic models that address cost as well as time.

We then discuss probabilistic time and cost-time representations, relating the material to the familiar program evaluation and review technique (PERT). This discussion brings us to an important sequence of ideas of generalized activity networks (GANs), the graphical evaluation and review technique (GERT), and the family of simulation languages known as GERT Simulation (GERTS).

Starting shortly after the introduction of activity network modeling and continuing to the present, development of the GANs/GERT/GERTS body of knowledge has made considerable progress. We believe that this body of work shows strong promise to support the CAIG's goal of widely applicable, effective modeling of the time and cost behavior of large development programs.

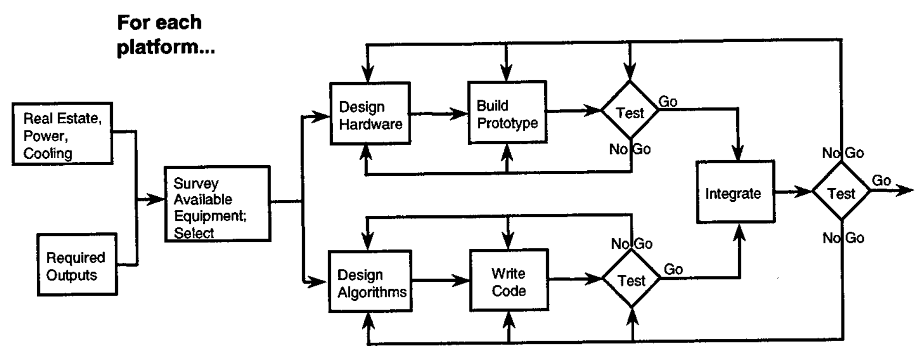
The section on probabilistic time and cost-time representations ends with a discussion of recent modeling work that is simpler and less general than GANs/GERTs, but, nevertheless, may offer a useful stepping-stone from PERT methods to those based on GANs.

Methods for Modeling Time and Cost of Development

The study of methods for modeling and analyzing the completion times of complex projects dates to the introduction of PERT in 1959.[2] Almost from the beginning of the ensuing four decades of research, workers in the field considered cost as well as time.[3] We will cite several recent parts of this knowledge. Some of the representations deal with time and cost deterministically, while others represent cost and time as dependent random variables. Computer applications packages are available to help analyze several of the representations.

Some of the more advanced of the available program models can capture the structures developed in this study for modeling development programs. For example, a representative structure of the development program for a subsystem of a complex product that emerged from our study of the electronics industry,[4] shown in Figure 2-5, could be represented by a GAN;[5] however, because of its feedback loops, CPM or PERT diagrams could not represent this structure.

Figure 2-5. Subproject Diagram



[2] Kelley, J.E. and M.E. Walker, "Critical Path Planning and Scheduling," *Proceedings Eastern Joint Computation Conference* **16**, 160–172, 1959.

[3] Kelley and Walker formulated a general cost-time tradeoff problem more than 40 years ago (Kelley, J.E. and M.E. Walker, *Critical Path Planning and Scheduling: An Introduction*, Mauchly Associates, Inc., Ambler, PA, 1959). Algorithms for solving the budget and deadline problems in polynomial time have been developed when the cost-time relations for each subprogram are deterministic, linear functions (Kelley, J.E., "Critical Path Planning and Scheduling: Mathematical Basis," *Operations Research* **9**, 296–320, 1961; Fulkerson, D.R., "A Network Flow Computation for Project Cost Curves," *Management Science* **7**, 167–178, 1961; Phillips, S.M. and M.I. Dessouky, "Solving the Project Time/Cost Tradeoff Problem Using the Minimal Cut Concept," *Management Science* **24**, 393–400), 1977.

[4] Belcher, G. and D. Lee, *Estimating Development Costs in the Defense Electronics Industry*, LMI Report PA805T2, 4-3, January 1999.

[5] Eisner, H., "A Generalized Network Approach to the Planning and Scheduling of a Research Program," *Operations Research* **10**, 115–125, 1962.

Deterministic Time and Time/Cost Representations

This section provides an introduction to the class of representations that we will consider by describing the classical CPM method and some of its extensions that include both cost and time.

Critical Path Method (CPM)

The CPM begins with a diagram of the project under study. We will use the term "activity-on-arc" (A-on-A) diagrams in this report. In an A-on-A diagram, arcs representing the actions of a subproject connect nodes representing events. The CPM cannot deal with loops (cycles). For example, the A-on-A diagram in Figure 2-6 represents the subprogram of Figure 2-5, without its feedback loops:

Figure 2-6. Example A-on-A Diagram

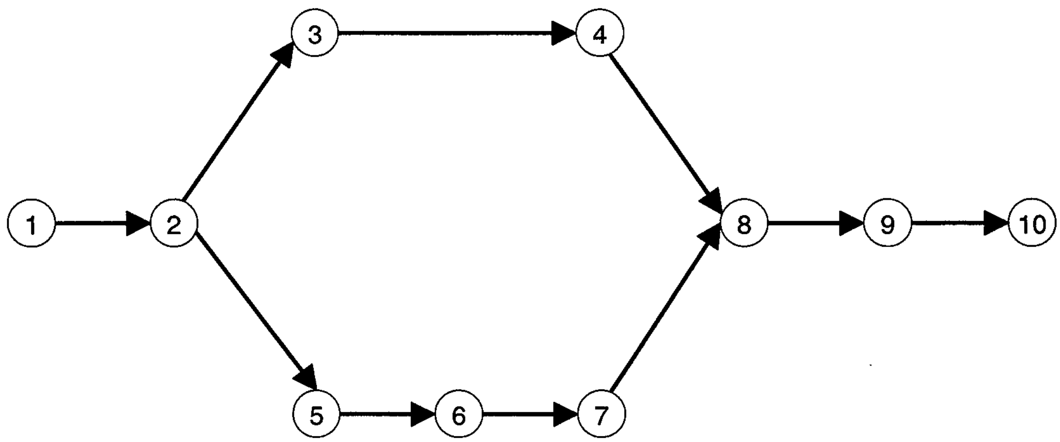


Figure 2-6 depicts an acyclic-directed graph, or activity network (AN). Here *directed* means that each arc can be traversed in only one way, as indicated by the arrows in Figure 2-6. *Acyclic* means that no sequence of arcs starting from a given node ever returns to that node. The acyclic character of an AN implies that it is always possible to number the nodes *1, 2,... n*, so that node *1* is the start of the project and node *n* is its completion. We also may describe any arc as going from node *i* to node *j*, with $i<j$.

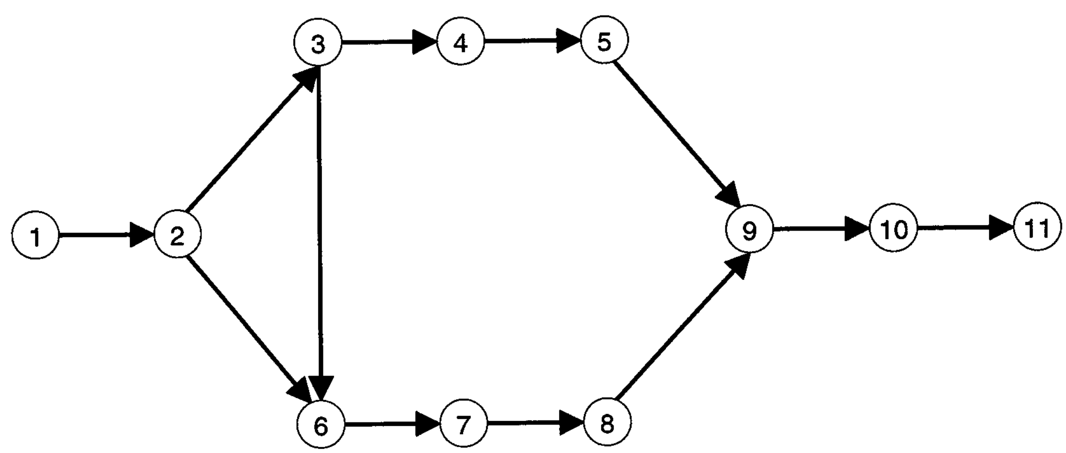
In Figure 2-6, the actions of the *survey available equipment, select* block in Figure 2-5 are represented by the arc *(1,2)*. The other arcs of Figure 2-6 represent the other blocks of Figure 2-5 in an obvious way. For example, arc *(2,5)* is *design algorithms*, arc *(3,4)* is *build hardware*, arc *(8,9)* is *integrate hardware and software*, and so on.

Given the times required to complete the activities on each arc, the objective of the CPM is to find the "critical path" (CP) (i.e., the longest-time path through the diagram). This is easy for the diagram in Figure 2-6. The CP is the arc sequence *(1,2), (2,3), (3,4), (4,8), (8,9), (9,10)* if the total time for the sequence *(2,3), (3,4), (4,8)* is larger than that for the sequence *(2,5), (5,6), (6,7), (7,8)*. If the time for *(2,3), (3,4), (4,8)* is smaller than that for *(2,5), (5,6), (6,7), (7,8)*, the CP is the sequence *(1,2), (2,5), (5,6), (6,7), (7,8), (8,9), (9,10)*. If the time for *(2,3), (3,4), (4,8)* is the same as that for *(2,5), (5,6), (6,7), (7,8)*, then both paths are equally critical. This last point is not insignificant. A careful practitioner will calculate the CP and a few of the next-to-critical paths to gain some idea of how readily events might upset an identification of a CP.

While CP analysis is obvious for the very simple case of Figure 2-6, it is by no means so for more complicated graphs representing real programs with some fidelity. We will say considerably more about which problems can be solved with reasonable effort in the subsequent section on computational complexity. For now we remark only that, because the task of determining an AN's CP can be reduced to a linear programming problem, that task, while challenging, is computationally feasible.

Sometimes it is desirable to show "partial precedence," in which certain parts of one task must be finished before other tasks can begin. ANs can accommodate such cases. Figure 2-7 shows an example, derived from Figure 2-6, in which a part of the *hardware design*, represented by arc *(2,3)*, must be finished before *software design* can begin.

Figure 2-7. AN Showing Dependence of One Task on Part of Another



Deterministic Time/Cost Representations

Effective means to treat some issues of cost as well as time for complex projects emerged many years ago. In a 1961 paper, Fulkerson[6] gave a network flow method for solving the set of linear programming problems that determines the cost of a project for any feasible time, when the project can be described by an AN, and when for each arc (i,j) the cost of completing the action represented by the arc varies linearly with the time to complete the action, between a "normal" completion time and a "crash" completion time.

Specifically, for each arc (i,j) of the AN, the three non-negative integers $a(i,j)$, $b(i,j)$, and $c(i,j)$ and a number $k(i,j)$ are given, and the cost $C(i,j)$ of completing the action represented by the arc in time $t(i,j)$ is displayed in Equation 2-2:

$$C(i,j) = k(i,j) - c(i,j)\,t(i,j), \qquad \text{[Eq. 2-2]}$$

for all $t(i,j)$ satisfying the conditions

$$a(i,j) \le t(i,j) \le b(i,j)\,. \qquad \text{[Eq. 2-3]}$$

Fulkerson's work gives the project's complete cost-time curve (which turns out to be piecewise linear and convex). It also turns out that, computationally, the linear cost-time tradeoff problem can be solved in polynomial time.[7] Others have revisited the linear cost-time formulation, giving improved algorithms.[8] Arsham developed a linear programming formulation and a new simplex-type solution algorithm for finding the changes in times for an AN's arcs that preserve an identified critical path.[9]

Whether or not problems related to a method of representing projects can be solved in polynomial time is important for the practical use of the method. We discuss this briefly in the next section.

Computational Complexity and the Practicality of Models; Budget and Deadline Problems

However elegant and complete a model or representation of a physical thing may be, it is not practical unless we can use it to solve relevant problems (at least in the sense of approximate solutions with explicit error bounds) in "reasonable"

[6] Fulkerson, D.R., "A Network Flow Computation for Project Cost Curves," *Management Science* **7**, 167–178, 1961.

[7] "Polynomial time" means that a computation's running time grows no faster than a constant multiple of a fixed power of n, such as $5n^2$ or $1066n^4$. These algorithms are said to be "easy." (Courant, R. and H. Robbins, *What is Mathematics?*, 2d ed., Oxford Press, Oxford, 510, 1996.)

[8] Phillips, S.M., and I. Dessouky, "Solving the Project Time/Cost Tradeoff Problem Using the Minimal Cut Concept," *Management Science* **24**, 393–400, 1977.

[9] Arsham, H., "Managing Project Activity Duration Uncertainties," *Journal of the Management Sciences* **21**, 111–122, 1993.

computing time. Discussing which problems have this desirable property is an important aspect of computer science. This aspect of a given problem is called the problem's computational complexity.

We can describe a problem's computational complexity by the way solution time varies with the problem's scale. A natural measure of the scale of problems connected with an AN is the number of arcs in it. This is the number of subprojects in the entire project, and often the number is denoted by J (the symbol is mnemonic if you think of subprojects as "jobs").

A problem is said to belong to *class P* if it can be solved in a time that is bounded by a polynomial function of its scale measure. Problems in class P are useful in practice, at least if the polynomial's degree is not too large.

Another class of problems, not smaller than class P, which may still be useful in practical cases, is the class of problems for which a candidate solution can be verified in polynomial time. This class is called *NP*. Obviously $P \subseteq NP$ (P is a subset of and is not larger than NP); it is one of the great unsolved problems in computer science to decide if $P = NP$.

For completeness, we note two other levels of computational complexity, *NP-hard* and *NP-complete*. NP-hard problems are a special class of problems with the property that, given a method for solving any one of them, any problem in NP can be solved with additional time that varies polynomially with scale. Thus, NP-hard problems are at least as computationally complex—as "hard"—as any problem in NP.

Some problems are known to be both NP-hard and NP. These problems make up a class of NP problems such that, if any one of them could be solved in polynomial time, then all NP problems could be so solved. This class is known as NP-complete problems.

Problems that are not in NP may not be useful for practical applications. For example, consider two problems, one of them in P, such that solution times vary as s^3, and another in neither P nor NP, such that even verifying a solution requires times varying like $\exp(s)$, where in both cases s is a scale measure for the problem. If, for each problem, the scale measure increases by an order of magnitude, the time required to solve the P problem goes up by no more than a thousand-fold. Thus, if the original problem required an hour to solve, the larger one would need a thousand hours. This is a significant increase in computer resources, but one which nevertheless might be handled by some combination of parallel processing and increased clock speeds, or even by brute force if one could wait about 42 days for the result.

But resources for the problem whose solution times increase exponentially would go up by a factor of $\exp(10)$, which is a bit over 22,000. That increase in resources might well exceed what one could do with parallel processing and increasing clock speed, and one probably would not be able to wait more than 2½ years for the result. With these ideas in mind, we will pay attention to the computational complexity of problems associated with the project representations we present here.

Two principal problems associated with any project seem particularly relevant to cost estimation. They are the budget problem and the deadline problem. The budget problem is, given an upper bound B on the project's cost, find a smallest time in which it can be completed. The deadline problem is, given an upper bound T on the project's completion time, find a smallest cost to execute it. A sequence of deadline problems, or a sequence of budget problems, maps out the variation of cost with completion time for an entire project. We expect, then, that a model must lead to computationally feasible budget or deadline problems, at least for approximations or simulations, if it is to be useful in cost estimating.

Discrete Time/Cost Representations

Because we can solve time-cost tradeoff problems on ANs in polynomial time when costs of activities vary linearly with completion time, we can conjecture that time-cost tradeoff problems on ANs for which the duration of each activity can be chosen only from a finite number of alternatives, each with its associated cost, also should be computationally tractable. Such discrete time-cost relations can approximate more complex relations between completion times of activities and their costs than linear ones.

In fact, that conjecture is wrong. The discrete time-cost tradeoff problem has been shown to be NP-hard.[10] This fact stands as a warning that the arithmetic of ANs for time and cost studies can pose challenges to using them in practice.

Nevertheless, recent work by Skutella shows that effective approximate solutions to discrete cost-time tradeoff problems are possible.[11] He gives an approximate algorithm for solving the discrete budget problem:

> *Given a fixed budget B, find the shortest realization whose cost does not exceed B.*

The algorithm executes in polynomial time, and produces a feasible solution whose completion time is not more than 3/2 of the shortest time. A realization of a project is a choice of the times for each subproject. The method limits the activities' duration to three discrete values. Skutella also presents polynomial-time algorithms producing feasible solutions whose values are within $O(\log L)$ of the optimum, when the possible durations are in the set $\{0, 1, 2,\ldots,L\}$.

[10] De, P., E.J. Dunne, J.B. Ghosh, and C.E. Wells, "Complexity of the Discrete Time-Cost Tradeoff Problem for Project Networks," *European Journal of Operations Research* **81**, 225–238, 1997.

[11] Skutella, M., "Approximation Algorithms for the Discrete Time-Cost Tradeoff Problem," *Mathematics of Operations Research* **23**, 909–929, 1998.

A sequence of budget problems sheds light on cost-time relations for the whole project. Skutella's helpful work indicates that we need not give up on modeling complex projects as ANs whose activity arcs have sets of discrete completion times with associated costs.

Probabilistic Time and Time/Cost Representations

In this section, for the sake of completeness and clarity of presentation, we begin with the classical PERT approach to ANs on which arc execution times are random variables. We also discuss recent literature, which seems to offer the possibility of achieving the CAIG's goals for project representations and models.

PERT

ANs for which the arcs' completion times are random variables are called *probabilistic activity networks* (PANs). Total time for each path through a PAN is a random variable, so in general there is no critical path. (There would be if total time for one path was greater than that for all others with probability *1*.) The critical path concept is generalized for PANs, to the idea of the criticality index. The criticality index of a path through a PAN is the probability that the path's duration is not less than the duration of every other path through the PAN. Criticality indices induce a partial ordering of the paths through a PAN, and generally serve to identify the "more critical" paths. One also speaks of the criticality index of an arc in a PAN. That is the sum of the criticality indices of all the paths through the network that contain the arc.

Execution time for a PAN is a random variable, obviously of considerable interest. Two reduction rules may allow one to generate the probability distribution function (pdf) of the total time to complete a project described by a PAN. First, the pdf of total time for arcs in series is the convolution of the two arcs' pdfs. And, second, the cumulative distribution function (cdf) of arcs in parallel is the product of the cdfs of the two arcs. Although the two basic reduction rules are simple, they quickly become computationally unwieldy for large networks.

Even determining criticality indices of all the activities in a PAN is computationally difficult. This motivates approximation methods and the use of simulations. Original PERT methods used beta distributions for completion times of subprojects (arcs), and sought approximations to the means and variances of total project times. The original PERT approximations have been heavily criticized.[12]

[12] See, for example, Elmaghraby, S., *Activity Networks: Project Planning and Control by Network Methods*, Wiley, New York, Chapter 4, Sections 1 and 3, 1977.

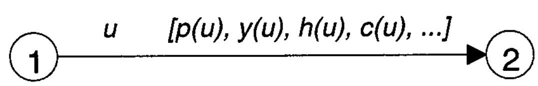
Other approximations have been better received. For example, Dodin contributed a computationally feasible method for approximating the criticality indices of the arcs in a PAN.[13]

Rather than seek exact or approximate pdfs for whole-network completion times of PERT ANs, we may apply simulations. While this approach is not without difficulty—determining how many examples to take, particularly for adequately determining "tails" of distributions must, as usual with simulations, be done with care—it is very helpful for dealing with large networks.

GANs, GERT, and GERTS

Recognizing that acyclic directed graphs are too limited to model all important features of projects—they cannot account for feedback, and every arc must be traversed—Elmaghraby introduced the concept of generalized activity network, or GAN, in 1964,[14] extending ideas presented by Eisner in 1962.[15] Closely following Elmaghraby's exposition,[16] we say that the basic element of a GAN is an arc connecting two nodes (Figure 2-8).

Figure 2-8. Basic GAN Element



Associated with the arc, labeled "*u*" in Figure 2-8, is an ordered set or vector of items: the probability, $p(u)$, that the arc will be executed; a pdf, $h(u)$, for the random variable $y(u)$, representing the arc's completion time; a cost function, $c(u)$, that may depend on $y(u)$ and so be a random variable; and any other parameters of interest.

Nodes in GANs represent events, which do or do not occur, given the condition of the arcs leading into them. As receivers, GAN nodes are of three types, shown in Figure 2-9.

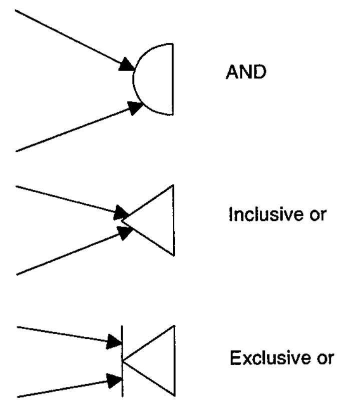
[13] Dodin, B.M., "Criticality Indices of the Activities in PERT Networks," North Carolina State University, *Operations Research Report* ARO1635218MA, February 1985.

[14] Elmaghraby, S., "An Algebra for the Analysis of Generalized Activity Networks," *Management Science* ***10***, 494–514, 1964.

[15] Eisner, H., "A Generalized Network Approach to the Planning and Scheduling of a Research Program," *Operations Research* ***10***, 115–125, 1962.

[16] Elmaghraby, S., *Activity Networks: Project Planning and Control by Network Methods*, Chapter 5.

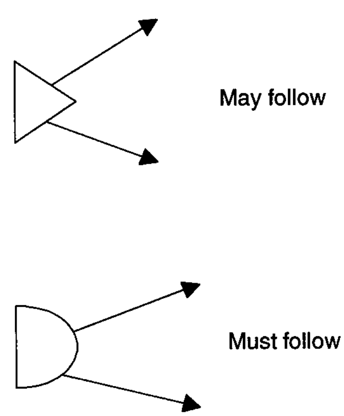
Figure 2-9. GAN Nodes as Receivers



The event represented by a GAN node that acts as an *AND* receiver is realized when, and only when, all arcs leading into it are realized. If the node acts as an *Inclusive or*, the event is realized when at least one of the arcs entering it is realized. The event of an *Exclusive-or* GAN node receiver is realized if one, and only one, of the arcs leading into it is realized.

As a transmitter, a GAN node may have one of the two behaviors shown in Figure 2-10.

Figure 2-10. GAN Nodes as Transmitters



When a GAN node is a *May follow* transmitter, the arcs coming from it are realized according to assigned probability distributions. One or more arcs may be realized with probability *1*, while precisely one member of each set of probabilistic arcs will be realized in accordance with a discrete probability distribution that is part of the specification of each such arc. For a *Must follow* transmitter, all arcs coming from it must be realized.

It can be shown that any GAN can, in principle, be replaced by an equivalent one in which all receivers are of the *Exclusive-or* type.[17] Analysis of GANs of that kind underlies a method for analyzing GANs, called the graphical evaluation and review technique (GERT). A GAN whose nodes all act as *Exclusive-or* receivers is called a GERT. The primary differences between a GAN and a PAN (for which PERT analysis applies) are that all arcs in a PAN will be executed, while some arcs in a GAN may never be executed, and a GAN can accommodate loops.

Obviously, GANs or GERTs provide quite flexible representations of projects. A GAN/GERT can be made that represents all the feedback loops shown in Figure 2-5. (Figure 2-11 shows an example of a GAN representing a version of Figure 2-5.) Also obviously, it seems unlikely that analytic results can be obtained for GANs except in very simple cases.

The earliest development of applications for GANs recognized this, and that led to development of a simulation language called GERT Simulation (GERTS). Development of GERTS has continued for more than 30 years, and applications are reported in the current literature.[18]

Elmaghraby 1990

We close this section with a brief description of some recent work that, while not as general as GAN/GERT, also may offer a useful modeling method. This material, attributed to S. Elmaghraby and his students, dates to 1990.[19] While presented as a means for firms to develop rational bids in competitive situations, it includes a project representation and computerized analysis methods that may be helpful in our present context.

The method applies to ANs, so we cannot consider feedback. One way to ameliorate this restriction would be to introduce "rework" arcs, with time and cost distributions appropriate for the parts of efforts undertaken in response to problems identified in testing. Each activity (arc) u has a set of discrete completion times, $y(u)$, with assigned probability distribution $\pi[y(u)]$. For each $y(u)$, there is a set of discrete cost outcomes $v_i[y(u)]$, with associated probability distribution $p(v_i)$.

This AN structure seems to offer a useful arena for modeling distributions of cost and time for MDAP development programs. Software has been developed for IBM-compatible PCs[20] that permits exploring the overall cost-time behavior of projects, and which can be expected to treat ANs with a few hundred arcs and

[17] Elmaghraby, S., *Activity Networks: Project Planning and Control by Network Methods*, 329 et seq.

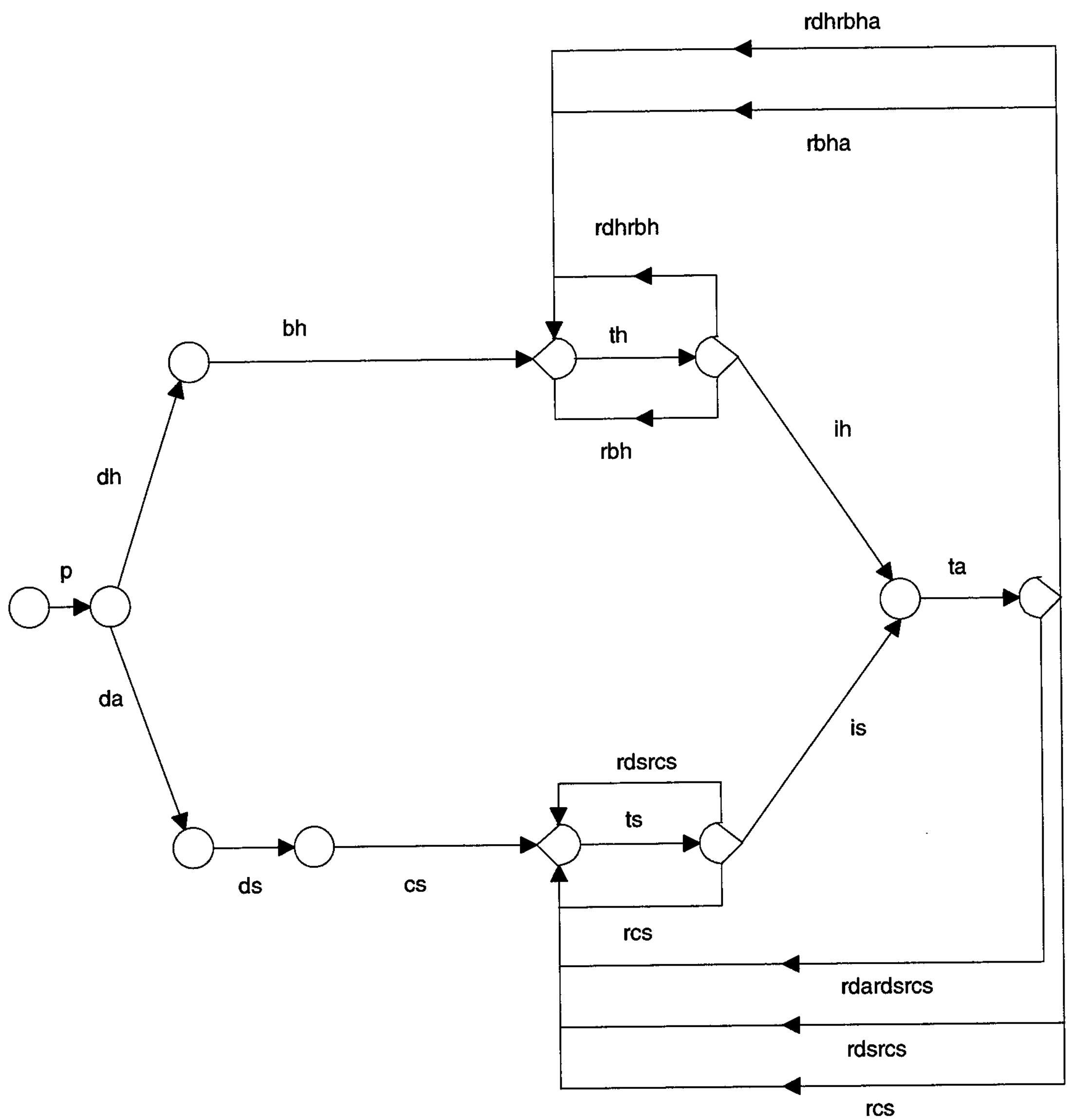
[18] Neumann, K. and W.G. Schneider, "Heuristic Algorithms for Job-Shop Scheduling Problems with Stochastic Precedence Constraints," *Technical Report*, Univerität Karlsruhe, June 1997.

[19] Elmaghraby, S., "Project Bidding Under Deterministic and Probabilistic Activity Durations," *European Journal of Operations Research* **49**, 14–34, 1990.

[20] Elmaghraby, S., and D. Michael, "Documentation of BIDNET: Project Bidding for CPM and PERT Activity Networks," *OR Report No. 221*, North Carolina State University, June 1988.

about a hundred nodes.[21] A GAN using this development approach is illustrated in Figure 2-11.

Figure 2-11. GAN for a Version of the Subproject Development of Figure 2-5



Here, rather than feed back through the original *design* and *build* processes, arcs are shown feeding back through *redesign* and *rebuild* processes. Thus, if the hardware fails its individual test, the program may require redesign and rebuild (arc *rdhrbh*) or only rebuilding (arc *rbh*). If the integrated system fails the all-up test (arc *ta*), the result could be redoing of hardware design and build, as well as algorithm design, software design, and software coding. This is represented by arcs *rdhrbha* and *rdardsrcs*, respectively. Less demanding rework after failing an

[21] Elmaghraby, S., personal communication to D. Lee, March 19, 2000.

all-up test is indicated by arcs such as *rdsrcs*, indicating software redesign and recoding.

PROGRAM EXECUTION SIMULATION

Using simulation to model complex projects to estimate the cost of such projects is intuitively palatable because systems analysts routinely use it to assist with systems analysis and management. A complex development program with its contributing activities can be viewed as a system, complete with stocks and flows we can simulate. For the DoD, OSD has mandated that simulation play a significant role in the acquisition of defense-related systems to cut costs, improve reliability, and bring systems into operation more rapidly.[22] So the use of simulation to support the resourcing and management of development programs is in keeping with OSD's philosophy.

Alan Christie of the Software Engineering Institute contends that, clearly, we can use simulation to predict the consequences of changing program requirements and to estimate and track project cost and schedule.

> Simulation can allow managers to make more accurate predictions about both the schedule and the accumulated costs associated with a project. This approach is inherently more accurate than costing models based on fits to historical data, since it accounts for the dynamics of the specific process. With regard to schedule, simulation can account for dependencies between tasks, finite capacity resources, and delays resulting from probable rework loops.[23]

With discrete-event simulation, we sample randomly from distributions describing time (schedule) and work (effort) required to perform program tasks. The output from such a model is distributions of schedule and effort to estimate the program's cost.

When we assess the use of discrete event simulation to model development program activities to accurately estimate program costs, we consider the following advantages and disadvantages.

Advantages

- Simulations can model complex, dynamic, real-world systems with several stochastic elements for which no analytical method is available.
- Performance of an existing system can be evaluated under different constraints or operating conditions.

[22] Christie, A., "Simulation—An Enabling Technology in Software Engineering," found at http://www.sei.cmu.edu/publications/articles/christie-apr1999/christie-apr1999.html

[23] Ibid.

- Alternative systems or operating policies can be compared.
- Experiments are reproducible and users can control experimentation conditions.
- Simulations allow for extreme compression of time, therefore, it is possible to study long-range effects in a short period of time.

Disadvantages

- Validation of such a model may be a problem; therefore, the predictive power of such a simulation may be suspect.
- Each run gives an estimate of true system performance. Statistical methods are required to give results more precision.
- Simulations of large systems can be expensive and time-consuming to develop and run.
- Large volumes of output data and attractive graphics often mask problems in the inherent assumptions.

Simulation Applications in Program Management: Lessons from Industry

In recent years, simulation models have been applied frequently to problems in project and process management in industries ranging from construction to aviation to electronics systems manufacturing, and more. In many cases, simulation proved successful because it enabled analysts to model more realistically the effects of change on the systems of interest. This in turn provided analysts and managers with valuable insights about the causes and effects of problems that otherwise would have been difficult, or perhaps impossible, to glean without simulation.

In this section, we discuss some of the lessons learned from applying simulation models to various organizations' problems. Although we found no cases that closely resemble the large-scale development programs to which we intend to apply our concept, we uncovered interesting uses of simulation that offer insights into the power and promise, as well as the problems, of this tool. We feel that these lessons are relevant to our current work.

Dealing with Uncertainty

Simulations have a distinct advantage over analytical methods in dealing with the uncertainty of, and variability in, program activities. A pharmaceutical company's experience in evaluating program-planning tools to help with staff planning serves as a good example. The company, which is involved in new product research and development, found that traditional spreadsheet analyses did not allow variability in program arrival patterns, program phase lengths, program resource needs, and program success. Consequently, the firm frequently had inaccurate views of programs, resources, and future revenues and costs. Static tools such as project management software, PERT, and CPM techniques did not account for the uncertainties in product development. Simulations did. The company found that it could better predict costs and revenues and get some sense of the variance in each.

A major management consulting firm also used discrete event simulation to solve its problems with program planning and realized several immediate benefits. The firm used flexible input mechanisms to capture planners' assumptions such as realistic program start dates. The firm was able over time to introduce variability such as arrivals, program phase lengths, attrition, and resource requirements into its analyses. The simulation had an easy to use format and organized results. The ability to vary arrivals, phase lengths, and attrition rates provides the company with useful insights. The firm uses the tool daily to make current and future staffing decisions and to test its business goals and reengineering efforts.[24]

Modeling Systemic Intricacies

Simulation consultants to the print/finish industry used simulations to help companies in the industry address the issue of how to get more volume through their systems in less time. They found static spreadsheet-based analyses often were misleading. In one case, these calculations erroneously showed there was enough machine time available to handle increased mail volume demands. Use of a simulation model, on the other hand, illustrated that the dynamics of the system would not allow the new mail volumes to be completed to meet the desired service level. The static calculations did not allow the needed in-depth analysis. Simulation helped the company identify where problems will happen and to address them in advance. In another case, simulation allowed a supervisor to test alternative strategies for machine assignments and to make the best selection for peak periods. To understand the implications of proposed system changes, we must include the dynamics of the process that simulation permits.[25]

A major electronic system manufacturer and a group of academic researchers jointly integrated discrete simulation with a popular costing software package. The joint simulation guided their decisions in making the transition from small volume, job-shop-like manufacturing to larger production-run volume manufacturing. The firm's cost package did not include variability in processing time, competition for scarce resources, or considerations for handling material accu-

[24] Grabau, M.R. and G.R. Clay, "Simulation Assisted Product Development Program Planning," *Proceedings of the 1999 Winter Simulation Conference*, found at www.wintersim.org.

[25] Benjamin, D., M. Curran, and T. Austin, "Simulation Case Studies in the Print/Finish Industry," *Proceedings of the 1998 Winter Simulation Conference*, found at www.wintersim.org.

rately. Other than reporting total cost and cost per part, the package lacked the capability to provide more detailed data regarding system performance that would help design the high-volume manufacturing facility. Discrete-event simulation was identified as a tool that could overcome the deficiencies of the cost package. The team concluded that the general approach might be applicable beyond its specific task.[26]

Adding Simulation to Ready-Made Program Management Tools

Academic researchers augmented a probabilistic CPM scheduling program with a simulation language, based on activity scanning and activity cycle diagrams, that was designed for construction projects. The researchers applied the tool to a highway construction project. Adding on simulation produced the flexibility and power to model uncertainty in the duration of activities as a true function of the state of the project. CPM allowed functions to be sampled from probability distributions. The parameters of these functions could include expressions or variables so that the researchers easily could model conditional and correlate distributions. The researchers included redefined cost expressions for the CPM program and created confidence intervals on project cost and duration.

The researchers used PERT network methodology to compute the cumulative probability of project completion. They also modeled the underlying process-level operations through concurrent simulation. In each replication, the simulation sampled the duration of activities and used this to perform the standard CPM calculations. The researchers concluded that the CPM integration illustrated the power of the simulation language in tackling their assignment. They also see teaching and research value in this CPM-simulation combination as a very useful probabilistic scheduling tool.[27]

Findings and Conclusions

Based on our research into the use of program execution modeling, we conclude that these modeling techniques would address many of the shortcomings of contemporary cost estimation tools. Modeling large, complex programs clearly calls for a technique akin to GERT, as described here. Like PERT and CPM, GERT networks have one source and at least one sink. But GERT networks can possess more general arc weights, several different types of nodes, and cycles to represent feedback (or, in our case, rework).[28] Though this technique typically is used as a machine scheduling tool for major manufacturing activities, its capabilities hold

[26] Harmonosky, C.M., J.L. Miller, et al., "Interfacing Simulation with Costing Software to Drive the Transformation from Prototype Manufacturing to High Volume Manufacturing," *Proceedings of the 1999 Winter Simulation Conference*, found at www.wintersim.org.

[27] Ioannou, P.G. and J.C. Martinez, "Project Scheduling Using State-Based Probabilistic Decision Networks," *Proceedings of the 1998 Winter Simulation Conference*, found at www.wintersim.org.

[28] Neumann and Schneider.

vast promise as a program execution model for deriving cost and schedule distributions.

We found that the intellectual capital represented by the GAN/GERT/GERTS knowledge provides a class of modeling and analysis techniques with excellent prospects for serving as the basis of greatly improved development cost estimates. This approach meets all the method-specific requirements listed in the "Overview" section of this chapter:

- Cost and completion times are treated as dependent random variables.
- Management's ability to assign resources may be represented by treating the probabilities that an arc is executed, and possibly also the dependence of cost on execution time, as decision variables.
- Precedence among subprojects is maintained.
- Project changes may be represented by probabilistic arcs without modifying the network; if necessary, a project's GAN can be modified with reasonable effort.
- Information gained as a project executes allows replacing probabilistic arcs with deterministic ones (one knows what happened), and this reduces dispersion in the project's time-cost distribution.
- Key questions, such as the project's overall cost-time distribution, can be answered with reasonable effort using available simulation methods.

These positive things said, it also must be said that the tasks of marshalling appropriate parts of that knowledge, and generating convenient applications packages to deal with DoD's specific needs, will not be simple ones. Could the entire DoD development community recognize the benefits of constructing and maintaining a GAN/GERT representation for a development program? The method certainly has current adherents, as illustrated by Neumann and Schneider. Would program offices willingly share with OSD the very considerable information about their operations that a project's GERT chart represents? We return to these points in Chapter 3.

We think it would be prudent to approach development of such a capability with some degree of caution. Particularly, we reiterate that systems of the magnitude of major defense development programs represent a significant level of computational complexity. The number of independent variables and parameters of interest one chooses to incorporate, of course, may further complicate this. Thus the methodology would require a powerful, and probably costly, programming application. Simulations may give helpful results in these complex cases.

Simulation, more specifically, discrete-event simulation, has been shown to enable program managers to gain valuable knowledge and insights about their programs. The prudent application of these simulations renders cost and schedule information and the uncertainties that surround each. We believe the combination of this tool with program execution modeling provides a powerful capability to program managers and cost estimators alike.

Chapter 3
The Way Ahead

We conclude from the research described in Chapter 2 that modeling the execution of complex development programs using discrete event simulation is both technically feasible and worth pursuing as a cost estimation method for DoD. At the same time, we do not want to understate the difficulties one should expect to encounter in developing a useful tool and attempting to implement the approach throughout DoD. The development of a useful tool for this application should not be viewed as an immediate-return goal of this research. The actual development of this approach into a practical capability will not be fast or easy. Indeed, the evolution of these concepts into a robust development cost estimation capability is in itself a significant development effort. This chapter outlines the vision for this new estimation tool and describes the effort we believe is required to bring the vision to fruition.

What Is Needed

The CAIG wants to field a generic tool that could be used by any development program. The tool would depend less on analogies and parametrics and more on the time and effort required to perform the specific tasks that make up a given program. We believe a tool can be developed and used effectively by CAIG analysts, acquisition program offices and cost analysis organizations alike to forecast the schedule and resultant costs of development. We believe that, for the CAIG's purpose, it is possible to develop cost-time diagrams (CTDs) of sufficient detail and complexity to capture relations among program activities, including significant allowance for feedback and cyclic activities. It also is possible to model effects of externalities like political pressures on funding, coupled with a simulation utility for generating answers to typical estimating questions.

We stress that this tool would *not* be a bottom-up cost estimating technique. Rather, it would be a standard framework for combining parametric models of program components made at appropriate levels, and for capturing the impacts of externalia not commonly considered in cost estimating. This tool could build on C-E and Pareto analyses used to find the true causes of development cost (see Appendix A). The factors and relationships highlighted in those analyses would be incorporated into this technique. Also, the probabilities associated with deciding if certain arcs are executed can be determined by considering factors such as political influence, technological state-of-the-art, and contractor incentives.

The developed system would be fielded at program offices for major acquisition programs, in OSD acquisition organizations, and in the CAIG. Regulations would require the system's use for an appropriate set of programs—say, all acquisition category (ACAT) I programs. We believe that the utility of the system to program offices would cause the offices to "build" and populate a CTD for their programs using acquisition guidelines and contractor provided data quite early in the project's life. Those offices would then share data and modeling results with service cost centers, the cost integrated product team (IPT), and the CAIG. As DoD implements more and more CTDs and shares information about the system throughout the acquisition community, the CTD system could become more effective over time.

The acquisition phases that concern the people responsible for spending DoD's development funds are Phases 0, I, and II. The first two phases of the process usually involve multiple commercial firms competing to receive DoD's funding to conduct further research and development and, eventually, manufacture products for the military. After the second phase (Phase I), DoD typically will "down select" or choose one of the remaining competitors' concepts. The primary objectives of Phase II, the "Engineering and Manufacturing Development" (EMD) phase, are to translate the most promising design approach into a stable, interoperable, producible, supportable, and cost-effective design; validate the manufacturing or production process; and demonstrate system capabilities through testing.

The CTD system we propose could be used to estimate Phase I costs; however, in some cases, so little will be known about a program's alternatives that the resulting estimate will show great dispersion. We believe that, at least initially, the system could be used best for Milestone II estimates, assuming estimators have access to source selection data and information.

We envision that a program's cost IPT, under CAIG leadership, would develop a CTD for a major development program as part of the preparation for its Milestone II review. The IPT would maintain the CTD during the entire EMD phase.

A successful CTD would need little change as the program executed. Maintenance would consist mainly of "pruning" arcs that, in the event, were not executed and making specific the costs and execution times of arcs that originally were random variables, as results become known. The CTD system should, nevertheless, provide for changing CTDs that did not have adequate provision for what actually happened as the program executed.

Dispersion of estimates produced from the CTD system will generally decrease as the program advances and actual values replace random variables. Extreme values will either contract toward expected values, or remain fixed. Expected values may either decrease or increase, depending on events.

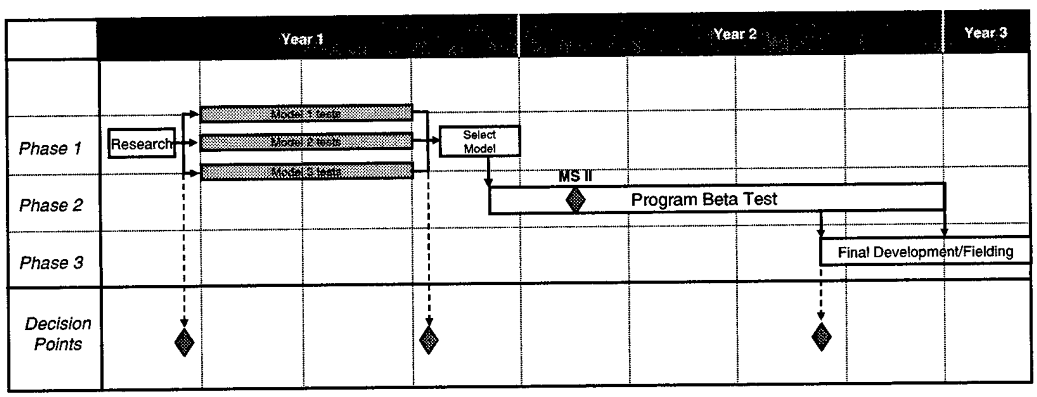
Concept Development

Taking our CTD concept from thought piece to fielded product will require the project management discipline of any significant software or information system development program. Development would start with literature and media searches, and possible selection, of COTS modeling capabilities from which to develop the system of choice. Developers would test modeling concepts using modest historical programs with well-known schedule and cost parameters. Development may progress such that testing first would be conducted with acyclic networks, then with a modeling approach of the type described by Elmaghraby, with rework arcs, and eventually with generalized ANs of the GERT class. We believe there are insights to be gained from each class of model that would contribute positively to the final product. Using progressively more complex models will add fidelity and robustness to the ultimate solution.

The Development Plan

This development effort would best be conducted in at least three phases as illustrated in Figure 3-1 (time scales denote *approximate* durations).

Figure 3-1. Program Timeline



- *Phase 1*: (Research, modeling, data collection, and prototyping) The development team would conduct search and evaluation to find robust, yet affordable project management models, one of which will form the framework for the prototype CTD system. If multiple promising candidate models are available, parallel testing with identical data sets may be used to evaluate each model's operations and results. The team would select a best tool based on CAIG-approved evaluation criteria. The research team then would apply the selected model to a set of straightforward historical programs to refine it and develop confidence in its ability to handle larger, more complex programs. Analysis and testing might include comparing model results to cost accumulation profiles from sources such as the Rayleigh Analyzer®. At this point, working through the CAIG, the team would identify an actual program in Phase I of its development cycle as the initial program to use the CTD system. The refined prototype would be used to develop and analyze a CTD of the identified program.

- *Phase 2*: (Concept demonstration) Working through the CAIG and the identified program's cost IPT, the team would develop a Milestone II estimate of schedule and total cost based on source selection data. In cooperation with the program's government project office, the CTD team would iterate the model to conduct sensitivity analyses throughout acquisition Phase II. The team also would maintain the CTD and modify it if necessary to capture funding or programmatic decisions.

- *Phase 3*: (Detailed development, production, and fielding) Taking results from the Phase 2 Beta test, the model would be refined and generalized for widest applicability. Considerable attention would be paid to developing the product's producibility. The model then would be disseminated to selected offices with appropriate documentation and support.

Model development and testing would be done under the supervision of a CAIG-led advisory group. Decision points would be included at logical stages in the program to allow for assessment and evaluation of results to date. The advisory group would make "Go-No Go" decisions regarding the remainder of the program based on its assessment of the potential to achieve a successful capability within the management parameters (i.e., budget and schedule).

Staff Work Required

- *Phase 1* (2–2.5 man-years) This phase would be particularly labor intensive. It would involve background research on models and program statistics to determine the most promising modeling and simulation approaches. The team would perform extensive data collection and statistical analysis to formulate the prototype CTD system. The team would test selected programs simultaneously with a least two modeling applications. This effort also may require some software engineering support to tailor packages to the specific task of modeling government product development activities.

- *Phase 2* (1.5–2 man-years) During this phase, the development team would conduct cost research required to develop a Milestone II estimate for the selected acquisition program using the selected modeling approach. The team would develop specific parameter estimates and operate and maintain the modeling system. Also, the team would provide modeling support to the program manager while drawing information and insights for CAIG (or advisory group) assessments.

- *Phase 3* (1–1.5 man-years) The team would develop a production estimation system; disseminate to ACAT I program offices and service cost centers; initiate, train, and certify operation of the installations.

Products

The primary product would be a fielded and supported estimation tool. Ultimately, the flexible simulation system would be used not only to estimate schedule and cost in accordance with acquisition policy, but also as a management tool for the program office. That would allow it to conduct sensitivity and risk analyses as the program progresses. It also could be used for impact analyses of budget cuts, requirement changes, and program slips.

In addition, several other useful products could come out of the successful development effort. Among these are

- a much improved DoD Cost Analysis Database with program execution parameters;
- an enhanced Contractor Cost Data Reporting (CCDR) templates and reporting mechanism; and
- a body of program execution and cost estimation research that may be applied to future development.

IMPORTANT CONSIDERATIONS

Verification, Validation, and Accreditation

Verification, Validation, and Accreditation (VV&A) will be a significant issue because the CAIG's vision calls for the system to be fielded across services and product sectors. As with any modeling project, the veracity of its results depend on how well the model represents what really happens. The developers will build on accepted project modeling principles and will apply those principles to projects or subprojects of limited scale and well-documented execution parameters.

A detailed VV&A plan must be developed to describe the procedures and standards for the developmental system. We expect the CAIG to be the accrediting authority. The accreditation procedures and requirements also would be developed and published in the VV&A plan.

Contract Cost Data Information

This cost estimating concept calls for more data, and more detailed data, than is reported routinely to the government for development programs. Current CCDR is insufficient to provide the required data to build and sustain this system.

Routine CCDR reporting is at Work Breakdown Structure (WBS) level 3; sublevel 3 reporting generally is required only for lower elements to address high risk, high value, or high technological interest. Each element's contribution to efficient decision making must be justified. (MIL–HDBK–881 identifies the first 3 levels of the program WBS and outlines the development of the contract WBS.)[1]

[1] "OSD CCDR Policy reaffirmed and updated," found at http://www.acq.osd.mil/pm/paperpres/ccdr.html.

For production programs, CCDRs must be submitted on delivery of each annual lot. Development contract reporting is defined by the needs of the program. At a minimum, reports need to be filed for major events (e.g., first flight, prototype fabrication), or before major milestone reviews. In general, quarterly or annual reporting does not meet these requirements.

The DoD cost analysis database is populated with data resulting from the CCDR process. DoD Instruction 5000.2–R provides mandatory guidance for the CCDR process. The CCDR Manual (DoD 5000.4–M–1, April 1999) implements that guidance and provides guidelines for contractors, program offices, and others. A major reengineering effort to update 5000.4–M–1, which had been virtually unchanged since 1973, was completed in 1999. The reengineering of the manual was accomplished through the formation of a CCDR Focus Group, established by the Office of the Secretary of Defense, Program Analysis and Evaluation Directorate (OSD [PA&E]) CCDR Project Office.[2]

The CCDR Focus Group continues to meet several times each year. The CCDR Project Office established a Software Metric Working Group (SMWG) to investigate how the DoD cost analysis community could obtain better software metrics—with the ultimate goal of improving software cost estimates. The SMWG proposed a revision to DoD 5000.2-R to require software metric reporting on all ACAT I programs. The SMWG proposal addressed only the data elements to be collected, not the process for collecting them. The metrics were limited to four basic measures that the Software Engineering Institute (SEI) recommends: effort, schedule, size, and quality. These data were extracted from the Cost Analysis Requirement Document Software Product Development Report, DD Form 2630. The revised form, DD Form 2630R, is half the size of the original form, and it is limited to directly measurable elements. The SMWG recommends the metrics be collected for each software release at each of three program phases: project initiation, contract award/project start, and product delivery. The SMWG intends for the data collected in the four SEI measurement areas to improve software cost estimates, allow analysts to study growth trends over a variety of projects, and be instrumental in the conduct of uncertainty analyses.

The current CCDR system is more directly applicable to production contracts and sometimes it is difficult to get meaningful data for development contracts. Improving hardware cost estimates and providing valuable data for simulation models could be achieved if similar measures, as described above for software development, were added to the CCDR process for hardware programs. OSD (PA&E) chairs the CCDR Focus Group. It may present to the group other modifications to CCDR requirements, if PA&E is satisfied that legitimate, necessary requirements exist for the data, and it would be cost-effective to collect and report those data.

[2] PA&E CCDR homepage found at http://ccdr.pae.osd.mil/.

We expect that a similar reengineering effort may be required to address development program data needs. Language may need to be added to the CCDR manual and DoD 5000.2-R to accommodate the new requirements. Table 3-1 illustrates additional data elements we feel would provide useful insights into cost drivers for development contracts.

Table 3-1. CCDR Information and Potential Additional Requirements

Report	Provided by the CCDR	Additional Development Data Element Examples
DD Form 1921	**By WBS reporting element** • Nonrecurring costs to date • Recurring costs to date • Nonrecurring costs at completion • Recurring costs at completion	**By WBS reporting element** *Same as shown, plus*: • Number of requirements • Number of interfaces • Redesign hours vs. planned hours • Rework effort • Test items needed • Unplanned expenses
DD Form 1921–1	**Engineering, Tooling, Quality, Manufacturing** • Direct labor hours, dollars • Overhead • Material • Other direct costs	**Hardware development, software development, integration, prototype, test assets, test labor** *Same as shown, plus*: • Redesign hours • Retest hours • Unplanned test assets • Total defects discovered • MTTD • MTBF
DD Form 1921–2	**By unit/lot accepted** • Direct quality control labor hours, dollars • Direct manufacturing labor hours, dollars • Material and purchased parts cost • Purchased equipment cost	Not applicable to development
DD Form 1921–3	**Plant-wide data** • Data on all systems being developed in the plant	Not applicable to development

Source: CCDR Manual (DoD 5000.4–M–1) found at http://ccdr.pae.osd.mil/manual/5000_4.pdf

Configuration Management

After the proposed system is fielded, it will be subject to numerous modifications to tailor its performance to the needs of diverse development programs and projects. Each of these alterations will serve as a data point from which systemic improvements will be sought. That evolutionary concept is one of the benefits of this concept. But this constant and widespread tailoring of the product makes configuration management critically important. The basic architecture and the current executable version must be centrally controlled. Configuration management responsibility should rest with the CAIG.

Customer Buy-In

Perhaps the greatest challenge facing this project is devising the incentives for development contractors and service and joint project offices to be open and to share information required to make the CTD system work. Realistic, effective program execution modeling requires that modelers have a clear understanding of the processes and activities required to develop products for DoD. All levels of the acquisition and cost analysis community must be given unprecedented access to materiel and manpower cost information, engineering and business management principles, and testing and evaluation assumptions and priorities. This likely will generate serious reservations among contractors and program managers.

Do we feel these reservations present insurmountable obstacles to the effective implementation of this concept? We do not. However, the case must be made emphatically that the precision of the estimates this system likely will provide will result in better allocation of resources and less waste due to unsupportable programs. Risk for both contractor and program office should be reduced if the system captures as many program activities and contingencies as possible. Disputes between contractor and government estimates will be minimized. Integration of cost and schedule with other program IPT concerns will be enhanced.

Declaration of these advantages must be reinforced, though, with strong acquisition resourcing policy that rewards programs for getting the Milestone II estimate right and heavily penalizes programs that do not.

SUMMARY

The evolution of the CTD cost estimation concept into a functional capability will require both a significant design and development project and widespread cultural change. The development effort amounts to 4½ to 6 man-years of effort encompassing in-depth research, data gathering, process modeling, and simulation. The cultural change required calls for projects and contractors to be extraordinarily open with cost data and internal procedures. It also requires the CAIG to assume the role of keeper of the keys to this new, powerful tool.

Bibliography

Ahmadi, Reza and Robert Wang. "Managing Development Risk in Product Design Processes," in *Operations Research*, Vol. 47, No. 2, March–April 1999.

Arsham, H. "Managing Project Activity Duration Uncertainties," *Journal of the Management Sciences* **21**, 1993.

Ballistic Missile Defense Organization Panel. *Reducing Risk in Ballistic Missile Defense Flight Test Programs,* 27 February 1998.

Belcher, Gerald and Dr. David Lee. *Estimating Development Costs in the Defense Electronics Industry,* Logistics Management Institute, January 1999.

Belcher, Gerald and John Dukovich. *Improved Methods for Estimating Development Cost: Cause-and-Effect Analysis,* Logistics Management Institute, January 2000.

Benjamin, D., Michelle Curran, and Tim Austin. "Simulation Case Studies in the Print/Finish Industry," *Proceedings of the 1998 Winter Simulation Conference*, found at www.wintersim.org.

Christie, A. "Simulation—An Enabling Technology in Software Engineering," found at http://www.sei.cmu.edu/publications/articles/christie-apr1999/christie-apr1999.html.

Clippinger, Anthony and Eric M. Gaier. *Characterizing Commercial Market Effects on Military Electronics Development Program Costs: An Analytical Framework,* Logistics Management Institute, September 1998.

Courant, Richard and Herbert Robbins. *What is Mathematics?*, 2d ed., Oxford Press, Oxford, 1996.

De, P., E.J. Dunne, J.B. Ghosh, and C.E. Wells. "Complexity of the Discrete Time-Cost Tradeoff Problem for Project Networks," *European Journal of Operations Research* **81**, 1997.

Dodin, B.M. "Criticality Indices of the Activities in PERT Networks," North Carolina State University, *Operations Research Report* ARO1635218MA, February 1985.

Eisner, H. "A Generalized Network Approach to the Planning and Scheduling of a Research Program," *Operations Research* **10**, 1962.

Elmaghraby, S., "An Algebra for the Analysis of Generalized Activity Networks," *Management Science* **10**, 1964.

---. *Activity Networks: Project Planning and Control by Network Methods*, Wiley, New York, 1977.

---. "Project Bidding Under Deterministic and Probabilistic Activity Durations," *European Journal of Operations Research* **49**, 1990.

Elmaghraby, S. and D. Michael. "Documentation of BIDNET: Project Bidding for CPM and PERT Activity Networks," *OR Report No. 221*, North Carolina State University, June 1988.

Fulkerson, D.R. "A Network Flow Computation for Project Cost Curves," *Management Science* **7**, 1961.

Grabau, Mark R. and Gregory R. Clay. "Simulation Assisted Product Development Program Planning," *Proceedings of the 1999 Winter Simulation Conference*, found at www.wintersim.org.

Harmonosky, Catherine M., Jennifer L. Miller, et al. "Interfacing Simulation with Costing Software to Drive the Transformation from Prototype Manufacturing To High Volume Manufacturing," *Proceedings of the 1999 Winter Simulation Conference*, found at www.wintersim.org.

Interview, Colonel Patrick O'Reilly, THAAD Program Manager, November 1999.

Ioannou, Photios G. and Julio C. Martinez. "Project Scheduling Using State-Based Probabilistic Decision Networks," *Proceedings of the 1998 Winter Simulation Conference*, found at www.wintersim.org.

John, Richard and Lawrence Kazense. *The Mechanics of Quality Processes,* ASQC Quality Press, Milwaukee, WI, 1993,

Kelley, J.E. and M.R. Walker. "Critical Path Planning and Scheduling," *Proceedings Eastern Joint Computation Conference **16***, 1959.

---. *Critical Path Planning and Scheduling: An Introduction*, Mauchly Associates, Inc., Ambler, PA, 1959.

Neumann, Klaus and Welf G. Schneider. "Heuristic Algorithms for Job-Shop Scheduling Problems with Stochastic Precedence Constraints," *Technical Report*, Univerität Karlsruhe, June 1997.

Office of the Secretary of Defense. OSD CCDR Policy found at http://www.acq.osd.mil/pm/paperpres/ccdr.html

---. OSD Earned Value Management System (EVMS) home page found at http://www.acq.osd.mil/pm/

Office of the Secretary of Defense, Program Analysis and Evaluation Directorate. Contractor Cost Data Reporting Manual (DoD 5000.4–M–1). found at http://ccdr.pae.osd.mil/manual/5000_4.pdf.

---. Contractor Cost Data Reporting homepage found at http://ccdr.pae.osd.mil/

Phillips, S.M., and I. Dessouky. "Solving the Project Time/Cost Tradeoff Problem Using the Minimal Cut Concept," *Management Science* **24**, 1977.

Process Improvement Guide, found at http://www.laafb.af.mil/SMC/MQ/qa/pigappa.htm

Skutella, M. "Approximation Algorithms for the Discrete Time-Cost Tradeoff Problem," *Mathematics of Operations Research* **23,** 1998.

United States General Accounting Office. "Missile Defense: THAAD Restructure Addresses Problems But Limits Early Capability," GAO/NSIAD-99-142, June 1999.

---. "Best Practices: Better Management of Technology Development Can Improve Weapon System Outcomes," GAO/NSIAD-99-162, July 1999.

University of Southern California Center for Software Engineering. "Design and Construction of Large Software Systems." found at http://www.sunset.usc.edu/classes/cs577b_98/risk_supplement.html

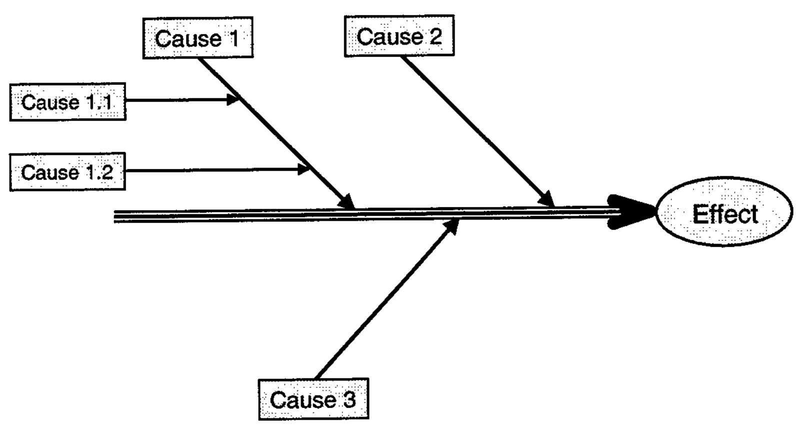
Appendix A
Cause-and-Effect Analysis

WHAT CAUSES DEVELOPMENT COST

The first and one of the biggest challenges facing research aimed at improving cost estimates is to determine what actually drives product development costs. What are the significant activities? What attributes characterize these activities? How do the activities contribute to what the government ultimately pays to develop a new product? Which of these is truly important? These are questions we must answer if we are to move toward a better understanding, and eventually a better estimation of development costs.

In this appendix, we attempt to accurately characterize development activities and costs for further analysis. We want to uncover the *causes* of development cost—the elements that make up development cost, in the general sense. One representational decision aid to facilitate the identification of the root causes of problems or issues is the cause-and-effect (C-E) diagram (shown in Figure A-1). Also referred to as an Ishikawa or fishbone diagram, the C-E diagram helps analysts to systematically examine the relationships between a given outcome and the factors influencing that outcome.[1]

Figure A-1. Basic Cause-and-Effect Diagram



The use of such a diagram not only allows analysts to sort out causes and organize relationships, but it also can act as a guide to data collection.[2] By the use of this

[1] Process Improvement Guide, found at http://www.laafb.af.mil/SMC/MQ/qa/pigappa.htm

[2] John, R. and L. Kazense, *The Mechanics of Quality Processes,* ASQC Quality Press, Milwaukee, WI, 207, 1993.

technique, we intend to lay bare the development process. Revealing the roots of the process will help us better model that process.

During this phase, we use anecdotal experiences from missile defense interceptor development programs to help build and illustrate the use of this model. We chose this particular sector because the products are highly sophisticated and uniquely military. We want to understand what, specific to the defense development environment, drives these developments. We look most closely at two key DoD missile defense interceptor programs—Theater High Altitude Area Defense (THAAD) and Patriot Advanced Capability-3 (PAC-3)—and we will refer to aspects of a third, the National Missile Defense (NMD) program.

The Causes of Development Cost

We use the C-E diagramming construct to model the processes and drivers that contribute to development cost. For our purposes, this method views ultimate cost to the government as an *effect* and strives to get at the root *causes* of cost; in an iterative fashion, to determine the "causes of the causes." With each additional level of detail, we can get closer to the roots of development cost. This method lends itself to detailed description of very complex problems, and it is not handicapped by the requirement that all its inputs be quantitative. The method mainly used to develop a C-E diagram is to survey many people or brainstorm with those who bring varying perspectives to the analysis.

We define "development cost" in this analysis as only the actual realized cost to the government for the development phases of the acquisition program. Using the traditional acquisition model, the relevant phases are Phase 0 (concept exploration) through Phase II (engineering and manufacturing development). We do not concern ourselves with the cost of activities or materiel to the contractor, which may bear little resemblance to the cost passed along to the government.

Approach

Following is our approach for this phase of the research:

- *Specify the problem and determine the major categories of factors influencing the problem.* We specify development cost as the problem or *effect* that we want to analyze. Later paragraphs summarize how we derive the major categories for our basic model.

- *Identify major factors and subfactors.* Ultimately, we are interested in predicting development costs; however, at this stage, we are concerned with what actually determines development costs. Our analysis approach to the problem of determining cause-and-effect is not to think in terms of *forecasting* what *will* happen, but rather to *record* what actually *did* happen. Essentially, we use 20/20 hindsight to describe what the contribution of each of the factors was for the programs of interest. This allows us to say, *with certainty*, what goes into development and development costs for these systems.

- *Identify and prioritize the significant roots of development cost.* We will trace promising factors to their sources, or roots. Using Pareto analysis or a similar technique, we will find and assign priority to the most influential roots for further quantitative analysis.

Caveats

The aim of this research is to derive factors and relationships that may be used universally to characterize the costs of various programs. We recognize that factors derived here may be logical only to the degree that the calibrating programs and the new program have much in common. The factors and estimating methods will require testing beyond the missile defense sector to verify their applicability to other developments.

The missile defense sector is unique in that there is no real civil analog. In our previous research, we found that in sectors where a military need seems likely to lead to profitable civilian sales, firms may be willing to bear some parts of development costs themselves, and thus keep the rights to the developed technology, rather than accept DoD payments for nonrecurring engineering.[3] Some costs will not be passed on to the government and, therefore, should not be reflected in the government cost estimate. This is a very different model than that for missile defense programs. This fact may limit the appropriateness of these findings to programs with limited application outside military markets.

THE BASIC CAUSE-AND-EFFECT MODEL

We analyzed the product development process and derived a first-cut C-E diagram. During the process, we hypothesized that development cost is a function of factors that can be cataloged in three major categories: the scope of the development project, the productivity of the project team, and economic or external factors that influence the development process and its costs. We initially assume that development cost is the product of these three major factors. This assumption may be modified later if the data so warrants.

[3] Belcher, G. and D. Lee, *Estimating Development Costs in the Defense Electronics Industry,* LMI Report PA805T2, 5-2, January 1999.

Displayed as an equation, the model looks like this:

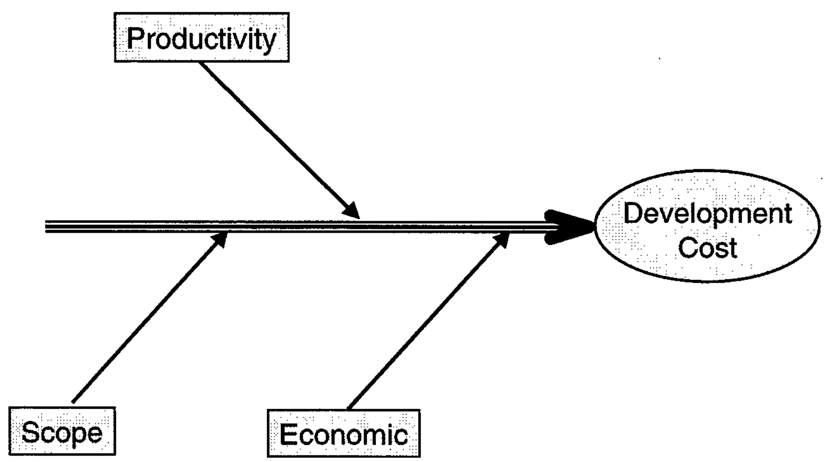
Development Cost = Scope × 1/Productivity × Economic Factors

where,

Development Cost is expressed in *$*,
Scope is expressed as *work performed*,
Productivity is expressed as *work performed/hour*, and
Economic Factors is expressed in *$/hour*

The basic C-E model yielded by using this framework is shown in Figure A-2.

Figure A-2. Basic Development Cost Cause-and-Effect Diagram



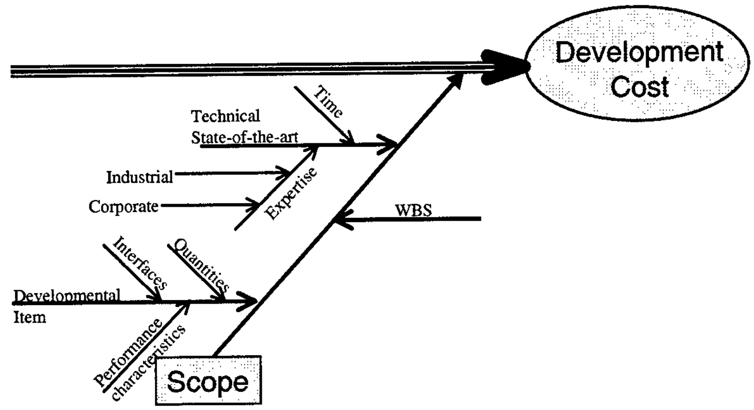
To expand this model, we then examined the PAC-3 and the THAAD programs to determine what makes up these major factors and how the subfactors manifest themselves in those programs.

MAJOR FACTORS AND SUBFACTORS

Project Scope

Scope simply describes what is being developed, or at least, what is *required* to be developed—the work to be performed. In general, several aspects of the requirement will affect what the development project will cost. Of course, the nature of the item under development is primary. How unique is it? How complex is it? What will it be required to do? How many items are required? Our discussion of project scope yielded the C-E diagram branch shown in Figure A-3.

Figure A-3. Project Scope



The Developmental Item

The bottom twig on this branch contains the specifications for the item to be developed. It specifies the primary requirements that a project manager will use to establish his project schedule, staffing, and budget. The most obvious of these requirements are the intended performance characteristics of the developmental item. These characteristics dictate whether the development can be based on a similar item or if it must start afresh. They also determine what developmental processes may be required and what technological hurdles the project is likely to face.

For the missile defense programs that are the focus of this research, the one characteristic that stands out as defining these products is the "hit-to-kill" requirement. The THAAD interceptor, the Patriot PAC-3 missile, and the NMD interceptor all must meet the very demanding requirements of "hitting a bullet with a bullet." This requirement dictates a high degree of precision engineering and very tight tolerances—things that drive development cost up. For those systems, the hit-to-kill requirement could be viewed as a root cause of development cost. It is the one aspect of these items to which most of the other scope-related factors are tied.

Interestingly, although program offices routinely cite "requirements creep" as a cause of development cost growth, that disease does not seem to have infected these missile programs. They did experience program changes that affected costs but, in the views of these program managers, those stemmed more from the development teams not fully *understanding* the complexity of the requirements from the outset. A result of this poor understanding was that development teams needed more tests and test articles than they anticipated, which drove development costs up. The development teams also underestimated the costs of integrating the many components and subassemblies of these very complex missile systems. In the contemporary environment, many larger programs are really the integration of smaller ones. Our findings here should serve as warning against taking the integration task lightly.

TECHNICAL STATE OF THE ART

Also depicted on the scope branch of the diagram is the *technological state of the art* for the item or component items under development. State of the art is a function of time—generally, it increases with time. A project engineer must be cognizant of the state of the relevant technology for the current project at any time, but also he must understand the technology trend, and how the project conforms to or varies from that trend over time. If a development project does not at least stay abreast of the state of the art trend, the development item runs the risk of being rendered obsolete before it is fielded. Staying abreast implies a factor of 1; falling behind the trend will cause a factor on cost greater than 1 (indicating cost growth).

State of the art also can apply to the level of expertise and sophistication of the development contractors and the industries in which they reside. If the industry in question, or the developing firm within that industry, does not keep pace with the technical state of the art, then the development project may be forced to catch-up or incur risk reduction costs. This really means more *work* must be done to keep the development on track with technology.

The THAAD Program Manager (PM) described the hit-to-kill mission of the new missiles as "on the frontiers of state of the art," but the actual items being developed were not considered overly sophisticated. The real challenge for these programs was to conduct the exacting engineering and integration required to make such precise weapons and to make them producible—common problems in high-technology fields. How these technology factors accumulate by integrating high tech components is still to be determined.

Relating cost to the relative expertise of the contractors is a difficult task for the missile programs because of the myriad subcontractors. For example, for THAAD, subcontractors performed approximately 80 percent of the design effort.[4] The experience level of the prime contractor may not always be reflected in a program's contributing subcontractors. Also, there was tremendous flux in the industries we researched; the major corporations merged and consolidated, sometimes creating corporate entities very different from their original pieces. The cost implications of these actions also are not immediately clear.

Obsolescence affected the cost of these missile programs in a number of ways. For THAAD, the Program Definition and Risk Reduction (PDRR) phase took considerably longer than originally planned. As the architecture standards for software changed over time, the software compilers for the program's many components became obsolete, adding cost. Similarly, the PAC-3 program suffered staffing cost increases because its software was written in the Ada programming

[4] Interview with Colonel Patrick O'Reilly, THAAD Program Manager, 16 November 1999.

language, which became all but obsolete during the program's progress. The Ada requirement made retention of software programmers difficult.

Work Breakdown Structure

The scope branch on the diagram also includes the processes and activities needed to meet the development requirements. These activities normally are captured in the *Work Breakdown Structure (WBS)* elements for the project. If the project builds from a legacy item, then no development activity, or perhaps only integration activity, is necessary. If the item is something that has never before been designed and built, then work will be required to design and develop the item in consonance with current technology trends or to "raise the bar" by moving the state of technology forward during the development effort. In this way, the true cost associated with the WBS can be viewed as a function of the increment of work required to close the gap between the state of the art technology and the requirement.

Complex missile defense programs demand extra testing and preparation for testing. According to a defense panel, "...the characteristics of HTK [hit-to-kill] programs demand increased emphasis on certain aspects of the paradigm (e.g., design margins, full qualification of components, careful analysis of critical functions and components, thorough ground end-to-end test, and so forth). The challenge of HTK [hit-to-kill] also warrants additional emphasis on HWIL [hardware-in-the-loop] testing and high-fidelity simulations."[5]

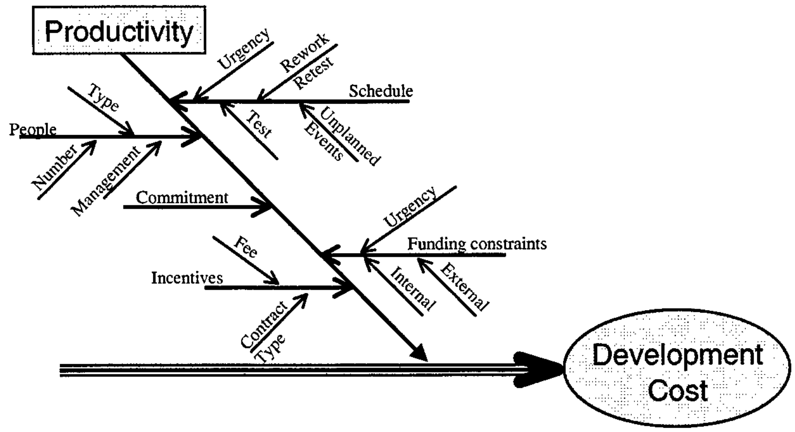
Programs such as THAAD and PAC-3 that call for large, complex hardware and software integration efforts often underestimate the amount of test, rework, and retest required to achieve success. Overly optimistic assumptions about the work required led both of these programs to incur additional costs because they had to make programmatic adjustments as the realities of program integration became apparent. The underestimation of activities also meant that the programs underestimated the requirements for test articles, a significant cost element in programs that require destructive testing.

Project Productivity Factors

The *productivity* branch of our model deals with those aspects of the project that determine *how* or *how well* the project team does its work. What resources will be used? What schedule must be maintained? What funding constraints are being applied? This branch is intended to establish those factors that determine how many increments of labor (i.e., manhours) it will take to complete the required development work (see Figure A-4).

[5] Report of BMDO Panel, *Reducing Risk in Ballistic Missile Defense Flight Test Programs,* 27 February 1998.

Figure A-4. Project Productivity Factors



PEOPLE

Perhaps the most visible determinants of project productivity are the quantity and quality of *people* making up the development team. In general, the cost-per-unit-time for work increases with the number of people employed and with the levels of expertise and experience those people bring to the development team. (Though using small teams or "young" teams do not necessarily mean cheaper development, employing the wrong people or the wrong number of people for the task generally will drive development costs up because it introduces inefficiencies into the development process.) Other significant aspects of the people or staffing twig include how the team is managed (how many interactions are required?) and the type (discipline or skill) of the members. There may be very high demand in the marketplace for some experts. The costs of using those experts, predictably, will be high relative to standard labor rates. (This applies for the use of subcontractors, too.) The supply-and-demand influence manifested itself in the PAC-3 program. As we mentioned earlier, the staff was unable to retain Ada programmers because Ada essentially became a programming language used exclusively for defense products. As programs and programmers shied away from Ada, the cost to the government of using Ada programmers increased.

These missile programs employ experienced staff but suffer high personnel turnover—another contribution to cost growth. The mergers and consolidations that took place in the defense industry in the 1990s caused a significant part of that turnover.

The missile programs we researched used a teaming approach to staff management common in these large-scale integration programs. This arrangement ties developers and development activities together for better or worse. If one team experiences problems, other teams may need to alter their pace to prevent a break in the work. For example, when the prime contractor's THAAD missile suffered test-related setbacks, the subcontractor's THAAD radar team had to slow its pace to preserve program integrity. Also, the large number of subcontractors employed by the prime were kept on retainer during delays because it would have been nearly impossible (and prohibitively costly) to release the subcontractors, and then restart the relationships when necessary.

SCHEDULE

We know from previous research that project *schedule* and scheduling problems ultimately contribute heavily to project cost. Indeed, the THAAD PM remarked, "The schedule *is* the driver." The number and type of test, rework, and retest activities, delays by contributing programs or in receiving component assets, and unplanned events all contribute to schedule-related project costs.

The combination of people and schedule factors directly results in cost. The efficiency with which management schedules people of varied skills and disciplines and other resources to the development process largely determines development cost. The number and type of management activities imposed on this process may control the costs of the process. Research into development risk in the design process indicates that "both overmanaged and undermanaged processes result in lengthy design lead time and high development cost."[6]

As with the WBS, schedule impact is often underestimated. A lengthy program clearly is a more expensive program. But a program whose schedule is originally too aggressive can take on risk that may result in rework and retest activities that lengthen programs and add significant cost. We believe this problem is fairly common because our previous research indicates that contractors and program managers often make optimistic schedule estimates to secure funding or to gain negotiating leverage with program contractors. The CAIG assessed the THAAD program's initial schedule to be too aggressive and, therefore, high risk. The program office estimated a 4-year development; the CAIG recommended 5 years. As of this writing, the program now is in its seventh year. It suffered schedule slips as a result mainly of flight test failures in 1996, 1997, and 1998. Failure to achieve optimistic schedule goals results in increased personnel costs and, in the cases of these missile programs, also requires increased material costs for more test articles.

Unplanned events are unforeseen occurrences that affect a project schedule. These occurrences often cause redirection of schedule or other resources away from the optimal allocation. These may be caused either by events external to the program, or by internal oversights or other shortcomings. PAC-3 testing scheduled at White Sands Missile Range in the summer of 1999 had to be delayed because of drought conditions in the area. This environmental concern affected the program's schedule, but early program planners and estimators could not reasonably have foreseen it.

To those schedule factors we add a measure of the *urgency* of the development effort that some projects experience. This perceived urgency might cause planners

[6] Ahmadi, R. and R. Wang, "Managing Development Risk in Product Design Processes," in *Operations Research*, Vol. 47, No. 2, March–April 1999.

to shorten project schedules artificially, thereby inducing risk that results in increased costs. These missile programs present classic examples. Congress mandated that the Army field PAC-3 in fiscal year 1998. The mandate applied extreme pressure on the schedule, which ultimately could not be met, and may have added significant risk and cost. THAAD, too, is saddled with a mandate to field a User Operational Evaluation System (UOES) before the objective system becomes operational. This capability, based on a perceived warfighting need stemming from the U.S. experience in *Operation Desert Storm*, became an end in itself and sidetracked the program from the original PDRR schedule. According to the report of a group empanelled to find ways for missile defense programs to reduce the risk in test programs, the UOES approach "...is inconsistent with the complexity of the task and has, thus far, not accelerated operational capability. Instead, the added risk has produced little discernible benefit and has actually delayed operational capability."[7] We realize that modeling or otherwise quantifying urgency may present considerable challenges for analysts. We assume this quality may somehow be represented by weightings or bias on the primary factors.

Funding Constraints

Next under productivity is funding, or more precisely, *funding constraints*. Without funding constraints, projects are free to obtain the best resources and to work to an optimum schedule. Clearly, it is rare that a program enjoys such freedom. More often, programs will suffer underfunding through imposition of constraints beyond their control or be required to make do if miscalculations or inefficiencies sap funds from the primary development effort. Reduced funding for the primary effort means fewer manhours are applied during a funding period than may be required for optimum development. Assuming the amount of work and the level of worker efficiency does not change, fewer hours for the same work mean a longer schedule (more funding periods required) and ultimately greater cost.

During the mid-1990s, the THAAD program suffered budget cuts amounting to about $2 billion. Program personnel believe that these cuts resulted in the contractor taking quality shortcuts to save money. The shortcuts may have been contributing factors in the failures of the first eight test flights. So funding reductions ultimately may have had the unintended effect of *increasing* the program's cost.

The PAC-3 program originally was fully funded in accordance with the CAIG's 1994 estimate. But monies had to be added when it became apparent that the development schedule was too compressed and would have to stretch. The program also has reprogrammed funds internally to address development contingencies. In essence, the program (really, all these programs) started as funding constrained.

[7] Report of BMDO Panel, *Reducing Risk in Ballistic Missile Defense Flight Test Programs*, February 27, 1998.

Commitment

Like urgency, *commitment* is an intangible quality. It takes the form of external priority and funding support and internal dedication of resources. Lack of either form of commitment may introduce risk and inefficiencies into the development process. For very high visibility programs, commitment in the form of priority and support is seen as key to program stability. Instability introduces risk, which adds cost. Neither of the missile programs suffered from lack of external priority. In fact, they are among the highest priority acquisition programs in DoD. Largely because of the high visibility of these programs, they are believed to have maintained a high degree of dedicated work and resources from government developers and contractors alike.

Incentives

Finally, any analysis of productivity where people are involved must include a discussion of *incentives*. A project team's productivity can be enhanced by the use of incentives, usually monetary. Projects can be prodded to higher levels of efficiency by the promise of incentive fees or by the type of contract vehicle the government uses. Incentives usually are linked to schedule or cost performance. The missile programs we are studying each used performance incentives to get desired results from their contractors. The PAC-3 program used incentives with its prime contractor by offering $25 million in performance awards. In the case of the THAAD program, an associate contractor agreement was struck between the government and the two major contractors. The agreement promised money to the contractors to induce them to work amicably together. The effect of these incentives on program cost is yet to be assessed.

The type of contract vehicle used may be viewed as an asset or a liability to the government. A General Accounting Office (GAO) assessment of the THAAD program's failures pointed to the type of contract vehicle the government used as contributing to the problem.

> The contract for developing the interceptor was a cost-plus-fixed-fee contract, a contract type that placed all of the program's financial risk on the government and [short of terminating the contract] did not include provisions that could be used to hold the contractor accountable for less than optimum performance.[8]

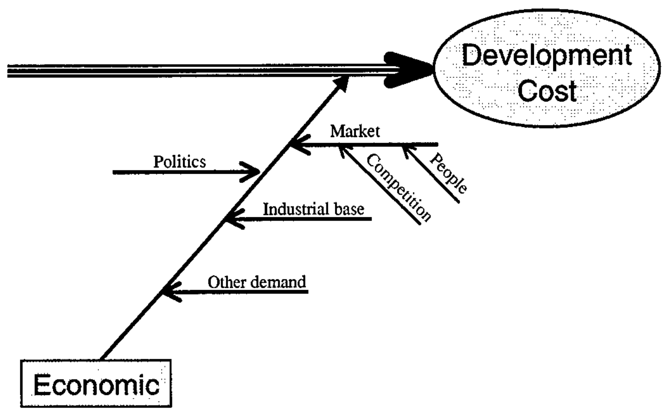
Of course, the government needs to conduct a cost/benefit or payoff analysis for each planned incentive to ensure that the expected benefits to the program exceed the costs of the incentives.

[8] United States General Accounting Office report, "Missile Defense: THAAD Restructure Addresses Problems But Limits Early Capability," GAO/NSIAD-99-142, 2, June 1999.

Economic Forces

The last major branch of our basic diagram deals with the *economic* conditions under which the program must be executed. This branch contains the significant externalities that make up the cost environment—those forces that help determine how available, and how expensive, time and labor will be (see Figure A-5).

Figure A-5. Economic Environment



The Market

First, the *marketplace* itself is a determinant of the environment. The program in question must enter the marketplace to obtain the appropriate people and other resources to perform its work. Because other contracts and other programs compete for the best people, the market exerts an upward force on personnel-related costs. At the same time, that marketplace exerts pressure to keep development costs down, at least for the purposes of competing for contracts and projects.

Because the THAAD program was so ambitious in terms of what the weapon system was asked to achieve, the number of serious competitors was limited. The large number of mergers and consolidations in the industry also meant that there was a tremendous turnover of expertise in the marketplace. The price of this expertise has remained very high. The length of the development program and its perceived risks, mainly because of test-failure-induced schedule slips, caused some subcontractors to lose interest and the program had to go looking for new subcontractors, again at increased cost.

Programs also will be affected by what we call the "cost of doing business." For example, the largest prime contractor for these missile programs had a relatively poor financial year as a corporation in 1999. There was speculation among the defense program managers that its overhead rates would go up and that the major acquisition programs would feel the upward pressure of these new rates. Another issue related to the use of subcontractors for components or assemblies. The

THAAD program office recounted how contractors at each stage of a development chain will add what they call a "wrap"—additional cost, similar to a value-added charge—to an item's cost. The program office estimated that the prime contractor passed along to the government costs that included wraps amounting to 40 to 50 percent of the original item cost. Recall that subcontractors performed about 80 percent of the design and development for THAAD; these wraps clearly could amount to considerable added cost to the government.

The Industrial Base

The status of the *industrial base* provides a measure of the state of the art in development and manufacturing that affects how well that base can support the development in question. The industrial base also may be focusing its talent and resources in other technologies or on other clientele. Either of these circumstances will drive costs up for our developments.

The industrial base relative to the number of major missile manufacturers is shrinking. Although this trend has slowed, now there is a limited number of firms capable of executing large-scale programs like THAAD and PAC-3. This Darwinian process has left a few, generally expensive, survivors. At the same time, these winners are very large, diverse corporations with wide interests, which split their focus into many pieces, and they have numerous priorities. Where a program falls on that list of priorities determines how the program's costs are affected.

Other Demand

We know from our previous research that *other demand* for the technology or product under development will affect either actual development *costs*, or what the DoD *pays* to develop a product, or both. In 1998, LMI conducted an economics-based assessment of the GPS industry and developed an analytical framework for understanding the effects the existence of a commercial market has on DoD development costs.[9] That work concluded that DoD's awareness of the commercial market interest and its implications for a defense development is necessary and may save the government money.

Though antimissile systems are inherently military, the contributing component technologies may have commercial demand. Firms contributing technologies that have other applications may be willing to accept some financial risk and save the government development costs.

Politics

The final factor on our diagram's economic branch is *politics*. This factor is meant to represent the various, mostly non-quantitative forces that affect programs for

[9] Clippinger, A. and E.M. Gaier, *Characterizing Commercial Market Effects on Military Electronics Development Program Costs: An Analytical Framework,* Logistics Management Institute, September 1998.

reasons having little to do with the programs themselves. The reasons can range from congressional, executive branch, or public pressure, to competing service, joint, or foreign development programs, to base realignment and closure (BRAC) issues. These forces may affect the progress of a program (for good or bad), regardless of the state of internal management or the availability of resources. Pressure from any of these quarters usually will result in increased costs for the development program.

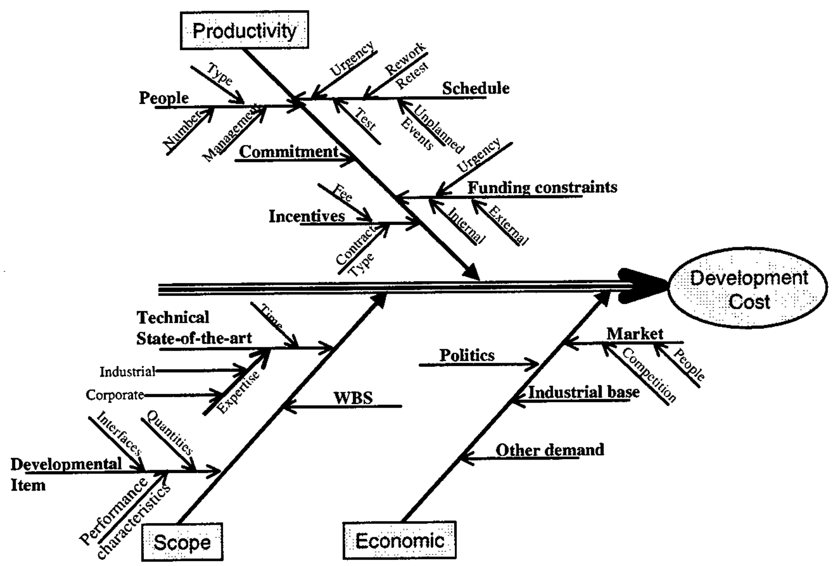
Again, the high price tags and the high visibility of the major missile defense programs almost guaranteed that they would be rife with political influence. The perceived critical need for an antimissile capability as soon as possible has affected all the major missile defense programs. Congress mandated that PAC-3 be fielded by FY98 while Congress and the Ballistic Missile Defense Organization (BMDO) both have had hands in restructuring the schedule and the funding profile of the THAAD program following the program's highly visible flight test failures. Because THAAD is being developed to have an exoatmospheric intercept capability, it also receives the scrutiny of organizations concerned with compliance with international treaties. NMD is among BMDO's highest priority programs and a congressional interest program. It, too, has been restructured largely because of legislative and executive branch pressures. Each of these factors tends to add cost to the programs.

THAAD acquisition also falls under the purview of both the Director of BMDO and the Army Acquisition Executive. These dual masters complicate the program's development process. The THAAD PM also commented about the effect of acquisition reform on his program. He believed that the relaxation of military specifications on developmental items allowed for the kind of quality problems the program experienced leading to the flight test failures.

FIRST-CUT C-E DIAGRAM

When we combine the factors we have described here, we get our first-cut model (see Figure A-6). This model forms the framework that guides our interviews with managers and engineers of legacy programs. We will use the results of the interviews and our research to update and refine our model. A logical next step might be to use data from those programs to begin the task of quantifying the causes and effects.

Figure A-6. First-Cut C-E Diagram

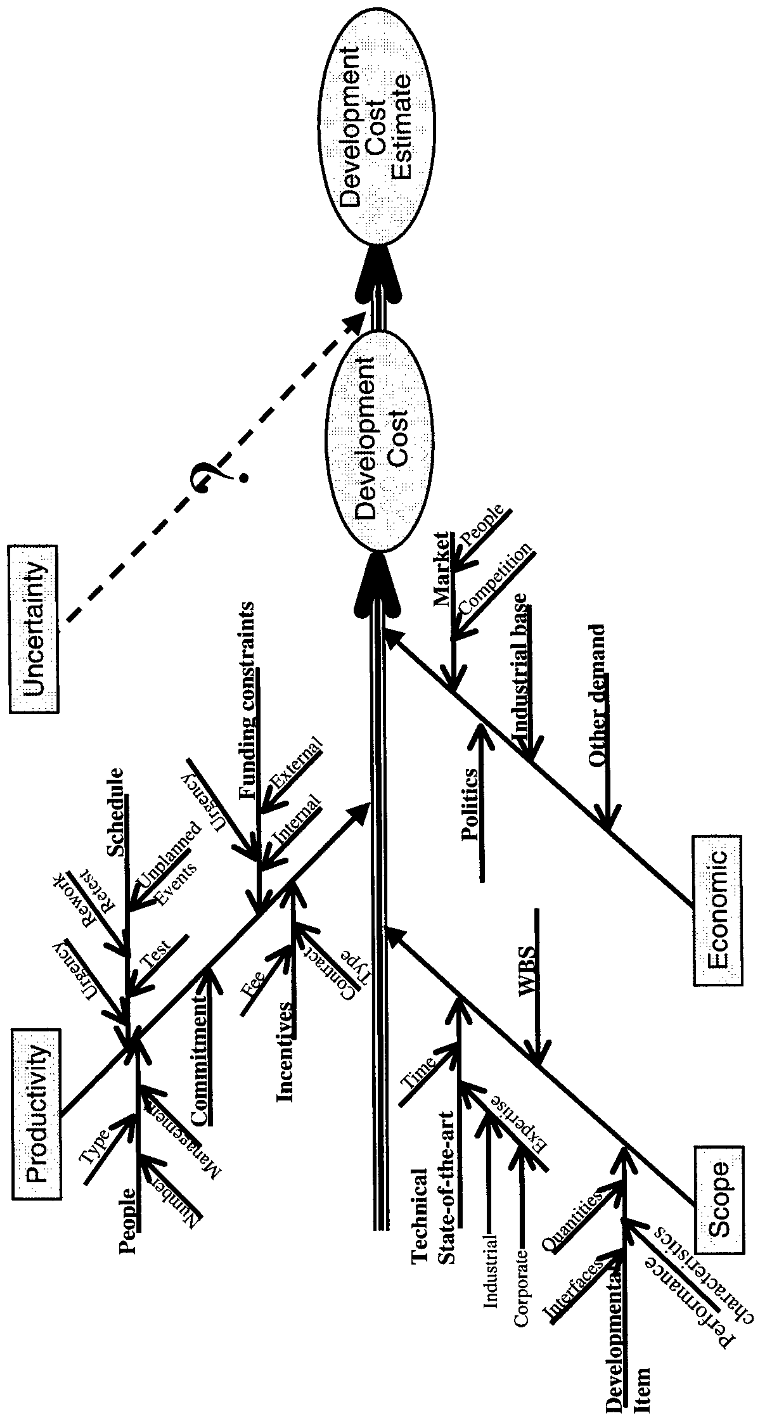


We have purposely defined our diagram in a shallow manner. It includes only those major factors that have visibility at very high levels (i.e., OSD level). Also, we did not want to assume the root causes in advance of the evidence. More interviews with subject matter experts may reveal significantly more detail. That detail should lead us to causes and relationships that will go far toward providing insights into the development process and, eventually, better cost estimates.

Uncertainty

As mentioned earlier in this paper, our approach would be to look back on programs to capture our factors with certainty. Clearly there will be variance in many of the factors. There will be *uncertainty* induced by technology and politics. There will be variability in the state of the market and in DoD acquisition funding. For these reasons, if this model is to be used as a *predictive* tool, we must account for this uncertainty. The subject of how to deal with uncertainty in development programs will be a major focus of analysis for the next phase of our research. For our first-cut C-E diagram, we simply will include a freestanding factor called "uncertainty" that, when applied to historical or analog development costs, allows us to make a development cost estimate or forecast (see Figure A-7).

Figure A-7. Full C-E Diagram with Uncertainty



We show the uncertainty factor on a par with scope, productivity factors, and economic factors. In reality, there will be uncertainty sprinkled throughout our model. After we collect enough data about the significant factors, we can apply probability distributions to them. We can propagate uncertainties through the model by sampling in the manner of an influence diagram.

Risk

Risk is a term that surfaced time and time again during our interviews with program office personnel. It was used to describe abstract feelings about the ability of the program to achieve a result. It also was used to describe concrete, quantifiable shortages of time or money. Perhaps the one aspect of program risk that can be captured and measured for our purposes is that which describes the relative status of requirements versus technology. A recent GAO report suggests a method for quantifying the readiness level of technology compared to the performance requirements of developmental weapon systems—a form of technology risk.[10] Program cost estimators may use these ratings as factors to determine the cost of this risk to their programs.

We would accept the definition of risk as the "possibility of an undesirable outcome, e.g., exceed budget, schedule overrun, delivering unsuitable product." Then the impact of risk is the product of the probability of the undesirable outcome and the cost of the undesirable outcome.[11] What is clear is that the effect of risk on development cost is not trivial. BMDO has developed a Cost-Risk Methodology that attempts to quantify cost risk as a fixed percentage of total program cost. Whether, where, and how risk should be portrayed in our model requires further analysis.

SUMMARY

The C-E model is an adequate representation of development cost's major factors for some classes of defense development programs. Given anecdotal information about a program, it enables analysts to display, systematically, the major factors that contribute to the cost of that program. Building C-E diagrams for historical programs shows what activities routinely are underestimated or missed altogether by program planners and cost estimators.

For the missile defense category of programs, we learned that a "stressing" performance requirement (i.e., hit-to-kill capability) may ultimately be viewed as a "root" of the program's development cost. We found also that planners routinely

[10] United States General Accounting Office report, "Best Practices: Better Management of Technology Development Can Improve Weapon System Outcomes," GAO/NSIAD-99-162, July 1999.

[11] USC Center for Software Engineering, Course description, CS577G, "Design and Construction of Large Software Systems,"
http://www.sunset.usc.edu/classes/cs577b_98/risk_supplement.html

underestimate the cost of integration and test activities. Finally, developers and estimators need very detailed understanding of system requirements. More research and analysis is required to develop the concept into a consistent framework that may be applied to different classes of development programs.

NEXT STEPS

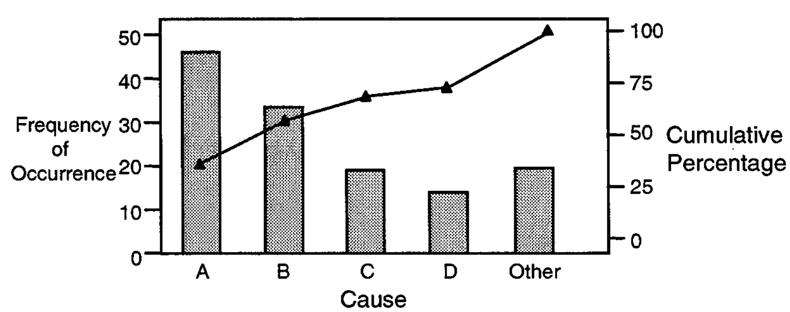
The process to this point has aimed at tearing development cost down to its elements so that we understand what those elements are and how they go together. It is this understanding that is fundamental to a good cost model. Future analysis must endeavor to understand the quantitative relationships among cost and its elements. The following paragraphs describe a process to move from our theory to a valid, working model of development cost.

Pareto Analysis

After we collect anecdotal information to help refine our basic model, we must find and prioritize the most influential factors for further quantitative analysis. It is those most influential factors that drive development cost. We must learn all we can about them.

Dr. Joseph Juran, a pioneer in the field of quality control, coined the term "the Pareto Principle" for his observation that most of a process's quality issues are the result of relatively few chronic problems. He named this principle after 19th Century economist Vilfredo Pareto's economic theories of the maldistribution of wealth. The principle says, "80 percent of the impact is caused by 20 percent of the problems." The Pareto chart, like the one shown in Figure A-8, combines a bar histogram with a cumulative line graph. The bars are placed from left to right in descending order, except for the catchall category on the right called "other." The cumulative line graph shows the percent contribution of all preceding bars. In our case, the chart will highlight the "vital few" factors or causes allowing us to target those for analysis and quantification. It will give some sense of the relative contribution of these factors to total development cost.

Figure A-8. Pareto Diagram



We can apply this Pareto analysis technique by surveying many programs using our derived C-E diagram and simply noting the frequencies associated with the various factors and subfactors. The factors with the highest frequencies of occurrence are given proportionately the most weight in whatever estimating model we use.

Bayesian Networks/Influence Diagrams

The weighted C-E diagram can be converted easily into a network with execution probabilities that we hypothesize. From these we could draw utility functions and create variables representing execution decisions. By definition, this is a Bayesian network or influence diagram. It would clarify the structure of the model for determining development cost. Next, we would analyze program data and cost reports for the programs of interest and analogous programs to capture the quantitative effects of the important factors we found through Pareto analysis. Deterministically modeling these effects should yield unfailingly accurate accountings of historical program costs.

We would conduct a thorough data analysis to establish the statistical descriptions of the various cost elements to include the degree of uncertainty surrounding them. Recognition of the uncertainty will allow us to assign parameters to the significant factors.

After describing the process completely, we would develop a stochastic model of development cost to account for the uncertainties in appropriate factors and processes. The resulting model then may be used as a tool to provide estimates of development costs, as well as consistent estimates of the uncertainties in the estimates. Chapter 2 of this report introduces some program modeling methods aimed at achieving that end.

Appendix B
Abbreviations

ACAT	Acquisition Category
A-on-A	activity-on-arc
AN	activity network
BMDO	Ballistic Missile Defense Organization
BRAC	base realignment and closure
CAIG	Cost Analysis Improvement Group
CCDR	Contractor Cost Data Reporting
cdf	cumulative distribution function
C-E	Cause and Effect
COTS	commercial-off-the-shelf
CP	critical path
CPM	Critical Path Method
CTD	cost-time diagram
DoD	Department of Defense
EMD	Engineering and Manufacturing Development
GAN	generalized activity network
GAO	General Accounting Office
GERT	graphical evaluation and review technique
GERTS	Graphical Evaluation and Review Technique Simulation
GPS	global positioning system
HTK	hit-to-kill
HWIL	hardware-in-the-loop
IPT	Integrated Process Team
LMI	Logistics Management Institute
MDAP	Major Defense Acquisition Program
NMD	National Missile Defense

OSD(PA&E)	Office of the Secretary of Defense, Program Analysis and Evaluation Directorate
PAC-3	Patriot Advanced Capability-3
PAN	probabilistic activity network
pdf	probability distribution function
PDRR	Program Definition and Risk Reduction
PERT	Program Evaluation Review Technique
PM	Program Manager
SEI	Software Engineering Institute
SMWG	Software Metrics Working Group
THAAD	Theater High Altitude Area Defense
UOES	User Operational Evaluation System
VV&A	Verification, Validation, and Accreditation
WBS	Work Breakdown Structure

REPORT DOCUMENTATION PAGE

Form Approved
OPM No.0704-0188

1. AGENCY USE ONLY ***(Leave Blank)***

2. REPORT DATE

May 00

3. REPORT TYPE AND DATES COVERED

Final

4. TITLE AND SUBTITLE

Better Development Cost Estimating Through Modeling and Simulation

5. FUNDING NUMBERS

C GS-23F-9737H

PE 0902198D

6. AUTHOR(S)

Gerald J. Belcher, John Dukovich, Dr. David Lee, Bobby Jackson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Logistics Management Institute
2000 Corporate Ridge
McLean, VA 22102-7805

8. PERFORMING ORGANIZATION REPORT NUMBER

LMI– PA903T1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of the Secretary of Defense, Director, Program Analysis and Evaluation
1800 OSD Pentagon, Room BE829
Washington, D.C. 20301-1800

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

A: Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

The Office of the Secretary of Defense (Program Analysis and Evaluation) tasked LMI to study how defense acquisition reform, advanced business processes, and new high technologies have affected how products are developed for DoD and what methods should be used to estimate the costs of those developments. In the first phase, LMI examined the defense electronics sector using Global Positioning System receivers as an analog. In that earlier research we found that extant methods and models were not well regarded and, therefore, were not used. This second phase was undertaken to study and address the weaknesses in the legacy approaches. Using military missile defense programs as the subject sector, LMI sought to better characterize the development process and better define the factors that contribute to development cost using cause-and-effect diagramming. With the findings of this investigation, LMI assessed the usefulness of some unconventional estimation techniques. Specifically, we assessed the use of simulations and program execution modeling with cost-time diagrams to forecast the costs and schedules of development programs. We developed a plan to implement this new system throughout DoD.

14. SUBJECT TERMS

Product development, development cost, cost estimation, cost models, discrete event simulation, cause-and-effect analysis, cost-time diagrams, GERT/GERTS

15. NUMBER OF PAGES

70

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL